

Контрольные вопросы к главе 1

1. Какую роль играет информация в живой природе? Какие действия с информацией осуществляют живые организмы?
2. Что считается носителем информации? Укажите традиционно используемые человеком носители информации.
3. Что называется знаниями? Укажите возможные способы хранения и передачи знаний.
4. Какую роль сыграли появление речи, письменности и книгопечатания в общественном развитии?
5. Укажите средства, применявшиеся человеком для обработки информации до появления компьютеров. В чем их особенности?
6. Охарактеризуйте ручной этап в развитии информатики.
7. Охарактеризуйте механический этап в развитии информатики.
8. В чем состояли идеи Ч. Бэббиджа?
9. Что общего между разностной и аналитической машинами Бэббиджа и чем они различаются?
10. Укажите основные причины и последствия информационного взрыва.
11. Охарактеризуйте электромеханический этап в развитии информатики.
12. Как определяется быстродействие компьютера? Что именно определяет эта величина?
13. Охарактеризуйте электронный этап в развитии информатики.
14. Какую роль сыграл Джон фон Нейман в становлении и развитии информатики?
15. Проследите связь между изменениями в средствах и способах хранения и обработки информации и общественным развитием.
16. Укажите основные предпосылки перехода к безбумажной информатике.
17. Опишите основные особенности персональных компьютеров.
18. Какую роль играет Интернет в человеческом обществе в начале XXI века?
19. Дайте определения терминов «пользователь», «информатика», «компьютеризация», «информатизация», «информационная технология».
20. Опишите особенности современных технологий работы с информацией.
21. Дайте определение науки информатики и опишите ее структуру.

Контрольные вопросы к главе 2

1. Дайте определение понятия «классификация».
2. Что может служить основанием для распределения по классификационным группам?
3. Перечислите возможные способы представления классификации.
4. Приведите примеры естественной и искусственной классификации. В чем принципиальное различие между этими видами классификаций?
5. Приведите примеры моделей, которые встречаются в повседневной жизни, профессиональной деятельности людей, науке и искусстве.

6. Сравните между собой когнитивную и содержательную модель, укажите их сходство и различия.
7. Дайте определение понятий «предметная область», «модель», «прототип», «моделирование».
8. Как происходит построение модели?
9. Охарактеризуйте основные этапы процесса моделирования.
10. Что означают термины «атрибут» и «абстрагирование»? Как они связаны друг с другом?
11. Какую роль в построении моделей играют гипотезы?
12. В чем заключается принцип подобия?
13. Опишите цикл построения и уточнения модели.
14. Перечислите и охарактеризуйте свойства, которыми должна обладать модель.
15. Перечислите и охарактеризуйте основные классы моделей.
16. Чем отличаются информационные модели от математических?
17. Что представляет собой вычислительный эксперимент?
18. Охарактеризуйте основные этапы компьютерного моделирования.
19. Приведите примеры систем, встречающихся в живой природе, окружающем мире, человеческом обществе.
20. Объясните термины «система», «системное свойство».
21. Охарактеризуйте основные свойства систем.
22. Какие общие методы работы с системами вам известны?
23. Что такое формальная система? Как задаются формальные системы? Приведите примеры формальных систем.

Контрольные вопросы к главе 3

1. Дайте бытовое, философское и используемое в информатике объяснение понятия «информация». Чем они отличаются друг от друга?
2. Чем вызвано существование большого количества трактовок понятия «информация»?
3. Чем вызвано существование неопределяемых понятий?
4. Чем понятие «информация» принципиально отличается от понятий «вещество» и «энергия»?
5. Что представляют собой источник информации, получатель информации, носитель информации, канал связи?
6. Охарактеризуйте основные аспекты понятия «информация».
7. Зачем понадобилось введение неопределяемого понятия «сообщение»?
8. Сравните между собой понятия «информация» и «сообщение».
9. Что означает термин «данные»?
10. Как связаны между собой сообщение и информация?

11. В чем проявляется практически важная сторона связи между сообщением и информацией?
12. Что требуется для того, чтобы извлечь информацию из сообщения?
13. Дайте определения терминов «информационная система», «система представлений» и «семантическая модель». Приведите важные для практики примеры информационных систем, систем представлений и семантических моделей.
14. Перечислите и охарактеризуйте основные свойства информации.
15. Перечислите и охарактеризуйте основные действия над сообщениями.
16. Чем отличается правило интерпретации сообщений от правила обработки сообщений?
17. Чем отличается диаграмма связей от коммутационной диаграммы?
18. Охарактеризуйте стандартно используемые носители сообщений. Чем различаются долговременные носители сообщений?
19. Что такое сигнал? Какие сигналы и какие их параметры чаще всего используются на практике для передачи сообщений?
20. Что представляет собой состояние объекта или явления? Чем отличаются друг от друга различные состояния? Приведите примеры различных состояний каких-либо объектов или явлений.
21. Дайте определение понятия «процесс» и приведите примеры различных процессов.
22. Охарактеризуйте информационные, стационарные и нестационарные, дискретные и непрерывные процессы. Какие из этих разновидностей процессов встречаются и используются в информатике?
23. Чем принципиально отличаются дискретные процессы от непрерывных?
24. Что представляет собой дискретное время? Что называется тактом? Почему в такт обязательно входит пауза?
25. Объясните две возможные трактовки понятия «отсчет».
26. Охарактеризуйте четыре возможных типа сигналов с точки зрения их непрерывности и дискретности.
27. Опишите особенности органов чувств человека, проявляющиеся в их взаимодействии с непрерывными и дискретными сигналами.
28. Назовите и охарактеризуйте возможные виды преобразования сообщений.
29. В чем состоит принципиальная особенность преобразования одних непрерывных сообщений в другие?
30. Для чего нужен переход от аналоговой формы сигналов к дискретной и как он происходит?
31. Как происходит развертка во времени?
32. Как осуществляется квантование сигнала?
33. Что такое ряд?
34. Для чего используется преобразование Фурье?
35. Сформулируете теорему Котельникова и объясните, в чем ее практическое значение.
36. Что такое частота дискретизации? Как она выбирается?

37. Чем отличается знак от символа? Что такое алфавит и чем он отличается от набора знаков? Приведите примеры алфавитов.
38. Какому требованию должен удовлетворять выбор алфавита?
39. Что представляет собой двоичный алфавит? Почему он считается наиболее важным для информатики?
40. Дайте определения терминов «двоичная цифра», «бит», «двоичный код», «машинный код».
41. Что представляет собой дискретное сообщение с точки зрения алфавита?
42. Что представляют собой цепочка над алфавитом и пустая цепочка?

Контрольные вопросы к главе 4

1. Охарактеризуйте основные задачи теории кодирования.
2. Сравните между собой задачи, решаемые теорией информации, и задачи, решаемые в теории кодирования.
3. Что определяет количество информации, содержащееся в некотором сообщении?
4. Опишите связь, существующую между информацией, неопределенностью и возможностью или необходимостью выбора.
5. Что называется информационной энтропией?
6. Дайте определения понятий «опыт» и «событие».
7. Чем случайные события отличаются от детерминированных? Чем вызвано такое различие? Приведите примеры случайных и детерминированных событий.
8. Дайте определение понятий «элементарный исход» и «пространство элементарных исходов». Какие требования предъявляются к элементарным исходам? Приведите примеры пространств элементарных исходов.
9. Чем события отличаются от элементарных исходов? Какие исходы считаются благоприятствующими, а какие — неблагоприятствующими для события?
10. Дайте определения понятий «невозможное событие» и «достоверное событие». Приведите примеры.
11. Чем совместные события отличаются от несовместных, зависимые от независимых? Приведите примеры и сравните между собой соответствующие пары событий.
12. Что представляют собой сумма и произведение событий. Приведите примеры.
13. Что называется вероятностью событий? Какими свойствами обладает вероятность?
14. Как вычисляется вероятность одиночного события? Выделите два случая вычислений. Какие события считаются равновероятными?
15. Что называется частотой события? Как связаны между собой частота и вероятность?
16. Что называется случайным числом? Приведите примеры случайных чисел.
17. Что такое математическое ожидание случайного числа? Как связаны между собой математическое ожидание и среднее арифметическое чисел?
18. Что понимается под энтропией опыта? Какими свойствами обладает энтропия?

19. Сравните между собой вероятность и энтропию. Что в этих понятиях общего и чем они различаются?
20. Какие формулы используются для расчета энтропии опыта?
21. Как связаны энтропия равновероятного и неравновероятного опытов?
22. Как связаны между собой энтропия и количество информации? Почему количество информации измеряется через энтропию?
23. Что такое бит? Приведите примеры вычислений количества информации.
24. Проследите связь между количеством информации и используемым для представления дискретного сообщения алфавитом.
25. Дайте определения понятий «шенноновское сообщение» и «шенноновский источник».
26. Какими способами определяется среднее количество информации, приходящееся на один знак алфавита?
27. Как рассчитать общее количество информации в поступившем сообщении?
28. Дайте определение понятий «слово», «код», «длина слова». Почему необходимо выделение слов в сообщениях?
29. Дайте определения понятий «первичный алфавит», «вторичный алфавит», «коддовая таблица».
30. Что понимается под операциями кодирования и декодирования? Каким свойством должны обладать эти операции, чтобы их можно было применять на практике?
31. Сравните между собой оптимальное и эффективное кодирование.
32. Что понимается под сжатием сообщений? Какие виды сжатия существуют? Сравните между собой два основных вида сжатия.
33. Что называется кодовой цепочкой? Что считается средней длиной кодовой цепочки?
34. Чем ограничена средняя длина кодовой цепочки при кодировании без потери количества информации?
35. Как рассчитывается теоретическое минимально возможное значение средней длины кодовой цепочки? От чего оно зависит?
36. Как рассчитывается фактическая длина кодовой цепочки для конкретного способа кодирования?
37. Сформулируйте теорему Шеннона. В чем ее фундаментальное значение для теории кодирования?
38. Что называется относительной избыточностью кода? Как ее можно уменьшить?
39. Что представляет собой энтропийное кодирование?
40. Опишите основные характеристики способов двоичного кодирования.
41. Сравните между собой равномерное и неравномерное кодирование.
42. Сравните между собой алфавитное и блочное кодирование.
43. Почему на практике применяется равномерное кодирование текстовых данных?
44. Опишите особенности и охарактеризуйте относительную избыточность азбуки Морзе.
45. Опишите особенности и охарактеризуйте относительную избыточность кода ASCII.

46. Что представляет собой базовая часть кодовой таблицы ASCII? Что считается ее расширением? Что понимается под кодовой страницей?
47. Дайте определение понятий «байт», «поле», «длина поля», «тетрада».
48. Как кратко записать двоичный код?
49. В чем состоит объемный способ определения количества информации в текстовом сообщении? Сравните между собой объемный и вероятностный способы подсчета количества информации.
50. Охарактеризуйте систему кодирования Юникод. Опишите ее внутреннюю структуру.
51. Сколько знаков может быть представлено в системе кодирования Юникод?
52. Что понимается под форматом данных? Охарактеризуйте стандартные текстовые форматы.

Контрольные вопросы и упражнения к главе 5

1. Что называется системой счисления? Какие системы счисления используются в компьютерах для кодирования числовой информации?
2. Что называется основанием системы счисления? Сформулируйте закон разложения по степеням основания системы счисления.
3. Найдите максимальное значение n -разрядного целого числа в системе счисления с основанием p .
4. Сформулируйте принцип получения изображений целых чисел в позиционных системах счисления.
5. Выпишите изображения целых чисел от 0 до 64_{10} в двоичной и шестнадцатеричной системах счисления.
6. Докажите, что наиболее эффективной системой счисления для представления данных в вычислительных системах является троичная. Почему тогда в компьютерах используется двоичная система счисления?
7. Сформулируйте правила перевода целых и дробных чисел из десятичной системы счисления в систему счисления с основанием p .
8. Переведите в двоичную и шестнадцатеричную системы счисления числа 185_{10} , 7929_{10} , -399_{10} .
9. Сформулируйте правило перевода чисел из системы счисления с основанием p в десятичную.
10. Переведите в десятичную систему счисления числа 10010001_2 , 110111_2 , 0101_2 , 91_{16} , $B8_{16}$, FAB_{16} , C_{16} .
11. Сформулируйте правила перехода между двоичной и шестнадцатеричной системами счисления.
12. Получите шестнадцатеричное представление для чисел 10010001_2 , 110111_2 , 0101_2 , 1011011110_2 , 10101_2 .
13. Получите двоичное представление для чисел $F48B_{16}$, $23F_{16}$, $9FE_{16}$, BC_{16} .
14. Сформулируйте правила выполнения сложения, вычитания и умножения в двоичной системе счисления.

15. Выполните следующие действия: $1010101,11101_2 + 11111,01101_2$, $1010101,11101_2 - 11111,01101_2$, $1001110111,1101_2 \cdot 10001,001_2$.
16. Сформулируйте правила выполнения сложения и вычитания в шестнадцатеричной системе счисления.
17. Выполните следующие действия: $AAFB89_{16} + 9DEFBA_{16}$, $FEDCBA_{16} + A0B3D9_{16}$, $4703B7_{16} - 3DEFBCA_{16}$, $69DEA0_{16} - 2FF_{EBA_{16}}$.
18. Какие форматы используются для представления числовых данных в компьютерах?
19. Чем с точки зрения информатики целые числа отличаются от вещественных?
20. Какие поля могут быть использованы для хранения целых чисел?
21. Опишите беззнаковое представление формата с фиксированной точкой.
22. Как связан диапазон представления беззнаковых целых чисел с длиной поля?
23. Сформулируйте порядок получения беззнакового кода заданного числа.
24. Получите беззнаковый код во всех допустимых полях памяти для чисел 121_{10} , 469_{10} , 78903_{10} .
25. Сформулируйте порядок определения числа по заданному беззнаковому коду.
26. Определите, каким числам соответствуют беззнаковые коды $C1_{16}$, $B4\ 06_{16}$, $15\ A0\ 02\ FE_{16}$.
27. Что называется разрядной сеткой?
28. Охарактеризуйте существующие системы знакового кодирования целых чисел.
29. Почему не применяется система кодирования со знаком?
30. Опишите принцип и способы получения дополнительного кода знаковых целых чисел.
31. Как связан диапазон представления знаковых целых чисел с длиной поля?
32. Сформулируйте порядок получения знакового кода для заданного целого числа.
33. Получите знаковый код во всех допустимых полях для чисел -283_{10} , $+199_{10}$, $-35\ 298_{10}$.
34. Опишите способ определения значения числа по его знаковому коду.
35. Определите, каким числам соответствуют знаковые коды $E2_{16}$, $AB\ 06_{16}$, $56\ 06_{16}$.
36. Как можно для заданного кода осуществить переход от поля одной длины к полю другой длины?
37. Опишите систему кодирования знакового кодирования со смещением. Где она применяется?
38. Опишите двоично-десятичный формат. Для чего он применяется? Охарактеризуйте существующие разновидности этого формата.
39. Сравните основную и экспоненциальную формы записи вещественных чисел.
40. Что представляют собой нормализованные числа?
41. Опишите особенности выполнения арифметических операций над нормализованными числами.
42. Сформулируйте общие принципы кодирования чисел в формате с плавающей точкой.
43. Как влияет выбор разрядной сетки на диапазон и точность представления вещественных чисел?
44. Что называется переполнением и антипереполнением? Когда они возникают?

45. Что определяет стандарт IEEE 754?
46. Охарактеризуйте особенности стандарта IEEE 754 при записи вещественных чисел в полях длиной 4 байта.
47. Получите для четырехбайтового поля значения максимального и минимального чисел. Какую точность имеют числа в этом поле?
48. Охарактеризуйте особенности стандарта IEEE 754 при записи вещественных чисел в полях длиной 8 байтов.
49. Получите для восьмибайтового поля значения максимального и минимального чисел. Какую точность имеют числа в этом поле?
50. Охарактеризуйте особенности стандарта IEEE 754 при записи вещественных чисел в полях длиной 10 байтов.
51. Получите для десятибайтового поля значения максимального и минимального чисел. Какую точность имеют числа в этом поле?
52. Опишите порядок получения кода заданного числа в формате с плавающей точкой.
53. Получите во всех допустимых полях памяти коды в формате с плавающей точкой для чисел $-22,5_{10}$, $78,35_{10}$, $-1045,168_{10}$.
54. Опишите порядок получения числа по его коду в формате с плавающей точкой.
55. Какие числа представлены следующими кодами в формате с плавающей точкой: E2 AB E4 02₁₆, C1 B4 00 00₁₆?
56. Как получаются специальные значения стандарта IEEE 754?
57. Охарактеризуйте назначение специальных значений стандарта IEEE 754?
58. Как происходит мягкое исчезновение порядка?
59. В каких ситуациях полезны денормализованные специальные значения? Что дает их использование?
60. В каких ситуациях полезны специальные значения бесконечность? Что дает их использование?
61. В каких ситуациях полезны специальные значения нечисло? Что дает их использование?
62. Чем отличается компьютерная арифметика от обычной? Приведите примеры различий.

Контрольные вопросы к главе 6

1. Охарактеризуйте различные аспекты понятия «алгоритм» и опишите сферы применения алгоритмов.
2. Почему возникла необходимость в точном определении понятия «алгоритм»? Чем нахождение решения задачи отличается от доказательства отсутствия решения?
3. Приведите различные объяснения понятия «алгоритм» на интуитивном уровне. Что в них общего?
4. Чем принципиально отличаются параллельные вычисления от обычных, последовательных?
5. Чем отличается описание алгоритма от его исполнения? Дайте определение понятия «исполнитель алгоритма».

6. Какую роль играют алгоритмы в интуитивно понимаемом смысле в человеческом обществе?
7. Приведите примеры алгоритмов, применяемых в различных сферах деятельности человека.
8. Сравните между собой понятия «алгоритм» и «программа».
9. Почему для практического применения в программировании алгоритмы должны обладать некоторым набором свойств? Кратко охарактеризуйте каждое из этих свойств.
10. Что подразумевается под дискретностью алгоритмов?
11. Охарактеризуйте все аспекты свойства конечности алгоритмов.
12. Что подразумевается под детерминированностью алгоритмов?
13. Сравните между собой детерминированные, вероятностные и недетерминированные алгоритмы. Приведите примеры алгоритмов каждого вида.
14. В чем заключается потенциальная выполнимость алгоритмов?
15. В каких случаях выполнение алгоритма завершается за конечное время и с получением требуемого результата?
16. Что требуется для понятности алгоритма? Кому он должен быть понятен?
17. В чем заключается свойство массовости алгоритма?
18. Что представляет собой машина Поста? Опишите ее устройство.
19. Чем отличается неограниченная лента от бесконечной?
20. Опишите возможности каретки машины Поста.
21. Дайте определения общих понятий «команда», «система команд», «программа». Конкретизируйте эти понятия для машины Поста.
22. Как связаны между собой машина Поста и ее программа. В чем эта связь аналогична связи между компьютерами и компьютерными программами и чем отличается от нее?
23. Что должно быть задано, чтобы машина Поста могла выполнить некоторую программу.
24. Опишите структуру команды машины Поста.
25. Опишите систему команд машины Поста.
26. Опишите порядок выполнения команды ветвления для машины Поста.
27. Каким требованиям должна удовлетворять программа машины Поста, чтобы обеспечивалось свойство потенциальной выполнимости? Как могут возникнуть невыполнимые команды машины Поста?
28. Какие ситуации могут возникнуть при выполнении программы машины Поста. Приведите примеры неудовлетворительных ситуаций.
29. Опишите постановку и основную идею решения задачи увеличения значения на единицу на машине Поста.
30. Для чего используется трассировка программы машины Поста?
31. Что понимается под отладкой и тестированием программы машины Поста?
32. Какова роль алгоритмизации в практическом плане?
33. Опишите известные вам способы задания алгоритмов в различных сферах человеческой деятельности.

34. Охарактеризуйте способ задания алгоритмов на естественном языке.
35. Что представляет собой машинный язык как способ задания алгоритмов? В чем его достоинства и недостатки?
36. Что представляет собой язык блок-схем? Охарактеризуйте их достоинства и недостатки. Для чего используются блок-схемы в настоящее время?
37. Опишите основные элементы, из которых состоят блок-схемы.
38. Что считается входом и выходом алгоритма?
39. Чем отличаются ассемблеры (автокоды) от машинных языков? В чем проявляются основные достоинства ассемблеров? В чем заключаются их недостатки?
40. Дайте определения понятий «алгоритмический язык», «транслятор», «трансляция».
41. Опишите общую схему применения алгоритмических языков при решении задач на компьютере.
42. Для каких целей могут применяться алгоритмические языки?
43. Чем различаются и чем похожи понятия «алгоритмический язык» и «язык программирования»?
44. Из каких обязательных элементов состоят системы языков программирования?
45. Сравните между собой используемые в информатике способы задания алгоритмов.
46. Перечислите и охарактеризуйте основные типы алгоритмов. Каким важным свойством они обладают?

Контрольные вопросы к главе 7

1. Дайте определения понятий «алгоритмический язык», «алфавит языка», «синтаксис языка», «семантика языка».
2. Какую роль в написании программы играют синтаксис и семантика языка?
3. Охарактеризуйте роль существующих в настоящее время видов семантики.
4. Для чего нужны метаязыки?
5. Охарактеризуйте язык БНФ. Как устроено определение в языке БНФ?
6. Для чего нужны комментарии? Как следует их формулировать? Опишите способы включения комментариев в текст алгоритма.
7. Опишите типичную структуру алфавита алгоритмического языка.
8. Что представляют собой ключевые слова в алгоритмическом языке с точки зрения их синтаксиса и с точки зрения использования?
9. Для чего в алгоритмах используются имена? Как правильно выбирать имена? Какие нотации связаны с выбором имен?
10. Какие виды данных используются в алгоритмах?
11. Какую роль в алгоритмах играют переменные? Охарактеризуйте переменную как конструкцию алгоритма. Какие переменные считаются неопределенными?
12. Как воспринимает процессор поступающие к нему на обработку коды?
13. Что дает концепция типа данных? Перечислите основные положения этой концепции.

14. Перечислите и охарактеризуйте базовые типы данных, встречающиеся в алгоритмических языках.
15. Опишите логический тип данных. Какую роль играют логические выражения в алгоритмах?
16. Как возникают данные целого типа? Опишите целый тип данных.
17. Как возникают данные вещественного типа? Опишите вещественный тип данных.
18. Сравните между собой целые и вещественные данные. В каких ситуациях необходимо проявлять осторожность, работая с вещественными данными?
19. Опишите символьный тип данных.
20. Чем отличаются простые типы данных от структурированных? Что представляет собой значение структурированного типа?
21. Какие структурированные типы данных используются чаще всего?
22. Что такое массив? Как выделить отдельный элемент массива? Что такое индекс?
23. Что такое запись? Как выделить отдельный элемент записи? Что представляет собой составное имя?
24. Чем отличается прямой доступ по индексу от прямого доступа по имени?
25. Какую роль играют описания в алгоритмах?
26. Как описываются переменные простых типов? Как описываются массивы?
27. Для чего нужны выражения? Как они устроены?
28. Чем отличается имя функции от ее указателя?
29. Сформулируйте основные правила записи выражений в алгоритмических языках.
30. Какая конструкция в алгоритмических языках считается отношением?
31. Какую роль играют операторы и управляющие конструкции в алгоритмах? Чем отличается оператор от управляющей конструкции?
32. В чем заключается действие присваивания? Опишите структуру оператора присваивания.
33. Опишите порядок выполнения оператора присваивания.
34. Сформулируйте основные правила использования оператора присваивания.
35. Сравните между собой оператор присваивания и отношение равенства. Чем они различаются принципиально и синтаксически?
36. Как организуется увеличение или уменьшение значения переменной на требуемую величину?
37. Что понимается под обменом данными в алгоритмах? Как организуется обмен данными?
38. Опишите структуру программы на языках Паскаль и Си.
39. Опишите примерный порядок работы над алгоритмом. Каким этапом завершается разработка алгоритма?

Контрольные вопросы к главе 8

1. Какие алгоритмы считаются линейными? Приведите примеры таких алгоритмов.
2. Что представляет собой управляющая конструкция «следование»?
3. Опишите общую схему построения линейных алгоритмов.
4. Сравните между собой варианты алгоритма решения задачи обмена значениями.
5. Сравните между собой варианты описания переменных для задачи решения системы линейных уравнений.
6. В чем заключается экономия вычислений?
7. Воспроизведите алгоритм решения задачи 8.2 и его трассировку.
8. В чем заключается тестирование алгоритма? Как оно проводится? Что необходимо для тестирования алгоритма?
9. Как происходит поиск ошибок в алгоритме? На что следует обращать внимание?
10. Какие алгоритмы считаются алгоритмами с ветвлением? Приведите примеры таких алгоритмов.
11. Опишите общую структуру алгоритмов с ветвлением.
12. Какие требования предъявляются к формулировке условий, от которых зависит организация ветвления?
13. Какие управляющие конструкции используются для записи алгоритмов с ветвлением? Нарисуйте соответствующие блок-схемы и запишите аналогичные операторы.
14. Как выполняется управляющая конструкция «ветвление»?
15. Когда используется полная форма ветвления, а когда сокращенная?
16. Воспроизведите алгоритм решения задачи 8.3 и его трассировку.
17. Опишите графические приемы, которые следует применять при записи алгоритмов на уровне алгоритмических языков.
18. Воспроизведите алгоритм решения задачи 8.4 и его трассировку.
19. В чем заключается стандартный способ построения новых алгоритмов?
20. В чем заключается опасность пропуска операторных скобок?
21. Опишите общую схему построения алгоритмов с большим количеством ветвей.
22. Как следует изображать блок-схему ветвления с большим количеством ветвей?
23. Как лучше контролировать правильность размещения ветвей в алгоритме?
24. Как подбирать тестовые варианты для проверки алгоритмов с ветвлениями?
25. Воспроизведите ход рассуждений при построении алгоритма решения задачи 8.5, его окончательный вид и трассировку.
26. Какие алгоритмы считаются циклическими? Приведите примеры таких алгоритмов.
27. Опишите общую структуру циклических алгоритмов.
28. Дайте определение терминов «тело цикла», «заголовок цикла», «условие повторения», «условие завершения», «инициализация цикла», «итерация цикла», «бесконечный цикл».
29. Опишите универсальную управляющую конструкцию, используемую для записи циклических алгоритмов. Нарисуйте соответствующие блок-схемы и запишите соответствующие операторы.

30. Как выполняется универсальная управляющая конструкция цикла?
31. Опишите общую схему построения циклов.
32. На что следует обратить внимание при проверке корректности построения цикла?
33. Опишите проверяемые в ходе тестирования периоды выполнения цикла и охарактеризуйте цель проверки каждого периода.
34. Какие последовательности называются рекуррентными? В чем их основное преимущество?
35. Почему невозможно построить линейный алгоритм решения задачи 8.6?
36. Воспроизведите алгоритм решения задачи 8.6, а также его трассировку, в том числе обратив внимание на особые случаи.
37. Как проявляется выход индекса за границы диапазона?
38. Сравните между собой две стандартные схемы построения цикла.
39. Где могут быть допущены ошибки при построении цикла?
40. Охарактеризуйте особенности цикла с параметром.
41. Какие действия в структуре цикла считаются организационными?
42. Что влечет за собой пропуск организационного действия?
43. Какими могут быть ошибки несогласованности организационных действий?

Контрольные вопросы и упражнения к главе 9

1. Сформулируйте задачу определения чисел Фибоначчи. Воспроизведите алгоритм ее решения, используя для построения цикла схему 1. Выполните трассировку алгоритма, в том числе для всех особых случаев.
2. Охарактеризуйте проблемы, которые возникают при использовании массивов для представления последовательностей, встречающихся в решаемых задачах.
3. Опишите способ представления последовательностей, основанный на простых переменных. В чем проявляются достоинства этого способа?
4. Объясните важнейший узел циклического алгоритма, который возникает при использовании простых переменных для представления последовательностей.
5. Воспроизведите алгоритм 8.6.1 решения задачи о рекуррентной последовательности, используя для построения цикла схему 1. Выполните трассировку алгоритма.
6. В чем проявляются недостатки применения простых переменных для работы с последовательностями? Почему возникающие при применении этого способа ошибки считаются опасными?
7. Опишите упрощенный способ решения задачи о рекуррентной последовательности, основанный на использовании одной простой переменной. В чем основная идея этого способа? Когда его можно применять?
8. Воспроизведите алгоритм 8.6.2 решения задачи о рекуррентной последовательности, используя для построения цикла схему 1. Выполните трассировку алгоритма.
9. В чем проявляется особенность способа, основанного на применении простых переменных, в случае рекуррентной зависимости от двух аргументов и от m аргументов? На что следует обращать внимание, выполняя переобозначения в таких алгоритмах?

10. Воспроизведите алгоритм 9.1.1 решения задачи о числах Фибоначчи, используя для построения цикла схему 1. Выполните трассировку алгоритма.
11. Объясните стандартный способ решения задач о накоплении сумм для случая, когда общий член суммы задан как функция его номера.
12. Как следует выделять входные, промежуточные и выходные величины при построении алгоритма?
13. Воспроизведите алгоритм 9.2 решения задачи о накоплении суммы, используя для построения цикла схему 0. Выполните трассировку алгоритма.
14. Опишите модификации стандартного способа накопления суммы.
15. Чем отличается алгоритм решения задач о накоплении произведения от алгоритма накопления сумм?
16. Воспроизведите алгоритм 9.3 решения задачи о накоплении произведения, используя для построения цикла схему 0. Выполните трассировку алгоритма.
17. Как вычисляются степени и факториалы?
18. Воспроизведите алгоритм 9.3.1 решения задачи о возведении в степень, используя для построения цикла схему 0. Выполните трассировку алгоритма.
19. Воспроизведите алгоритм 9.3.2 решения задачи о вычислении факториала, используя для построения цикла схему 0. Выполните трассировку алгоритма.
20. Что представляет собой двойной факториал? Чем алгоритм его вычисления отличается от алгоритма вычисления обычного факториала?
21. Воспроизведите алгоритм 9.3.3 решения задачи о вычислении двойного факториала, используя для построения цикла схему 0. Выполните трассировку алгоритма.
22. Что представляет собой скалярное произведение векторов? Как его можно вычислить?
23. Воспроизведите алгоритм 9.4 решения задачи о скалярном произведении векторов, используя для построения цикла схему 0. Выполните трассировку алгоритма.
24. Чем задача о вычислении суммы векторов принципиально отличается от задачи вычисления их скалярного произведения?
25. Воспроизведите алгоритм 9.5 решения задачи о сумме векторов, используя для построения цикла схему 0. Выполните трассировку алгоритма.
26. Что такое смешанное произведение векторов? Постройте и запишите алгоритм вычисления смешанного произведения векторов.
27. Что такое векторное произведение векторов? Постройте и запишите алгоритм вычисления векторного произведения векторов.
28. Сравните между собой прямой и эффективный методы вычисления значения полинома в точке.
29. Воспроизведите алгоритм 9.6 решения задачи о значении полинома, используя для построения цикла схему 0. Подберите несколько тестовых вариантов и выполните для них трассировку алгоритма.
30. Опишите основную идею алгоритма вычисления сумм (произведений), у которых элементы имеют рекуррентную связь. Как в данном случае происходит объединение алгоритмов?

31. Воспроизведите алгоритм 9.7 решения задачи о суммировании рекуррентных слагаемых. Подберите несколько тестовых вариантов и выполните для них трассировку алгоритма.
32. Как обнаружить рекуррентную связь между соседними слагаемыми?
33. Как организуется суммирование произвольных слагаемых? Сравните между собой два основных способа такого суммирования.
34. Воспроизведите алгоритмы 9.8, *a* и *б*, решения задачи о суммировании произвольных слагаемых. Подберите несколько тестовых вариантов и выполните для них трассировку алгоритмов.
35. Дайте определение понятий «ряд», «сумма ряда», «частичная сумма ряда». Какие ряды считаются сходящимися?
36. В чем состоит основная идея вычисления суммы ряда? Что считается остатком ряда?
37. Как на практике организуется вычисление суммы ряда? Как приближенно оценивается погрешность такого вычисления?
38. Воспроизведите алгоритм 9.9 решения задачи о суммировании ряда. Подберите несколько тестовых вариантов и выполните для них трассировку алгоритма.

Контрольные вопросы и упражнения к главе 10

1. Почему алгоритм решения задачи определения экстремального элемента, основанный на попарном сравнении элементов, неэффективен?
2. Опишите основную идею алгоритма определения экстремального элемента.
3. Воспроизведите алгоритм 10.1 определения экстремального элемента, используя для построения цикла схему 0. Выполните его трассировку.
4. В чем состоит типичная ошибка в алгоритме получения экстремального элемента массива?
5. Опишите особенности возможных модификаций алгоритма определения экстремального элемента массива.
6. Воспроизведите алгоритм 10.1, *a*, определения номера экстремального элемента, используя для построения цикла схему 0. Выполните трассировку алгоритма.
7. Каким образом в задаче определения экстремального элемента могут быть выбраны начальные значения переменной, служащей для фиксации результата?
8. Сформулируйте задачу поиска в наиболее общей постановке. Охарактеризуйте ее значение в информатике и других областях деятельности человека.
9. Что является основным ответом в задаче поиска? Какими могут быть дополнительные результаты ее решения?
10. Сформулируйте задачу поиска для случая массива. Какие ограничения накладываются в этой задаче?
11. Какой прием используется для фиксации результата в классическом алгоритме решения задачи поиска?
12. Какой способ решения задачи называется линейным поиском? Когда можно применять этот способ?

13. Как формируется условие повторения для цикла в классическом алгоритме решения задачи поиска?
14. Опишите со смысловой точки зрения варианты удовлетворения условия завершения цикла в задаче поиска.
15. Опишите особенности инициализации цикла в классическом алгоритме поиска.
16. Воспроизведите алгоритм 10.2 решения задачи поиска. Выполните трассировку алгоритма для нескольких различных массивов и искомым элементов.
17. Опишите особенности возможных модификаций классического алгоритма поиска.
18. Как осуществляется запоминание результатов в задаче поиска всех элементов, совпадающих с заданным?
19. Воспроизведите алгоритм 10.4 решения задачи поиска всех элементов, совпадающих с заданным. Выполните трассировку алгоритма для нескольких различных массивов и искомым элементов.
20. В чем состоит идея алгоритма быстрого поиска? Почему этот алгоритм эффективнее классического? Как определяется результат в алгоритме быстрого поиска?
21. Воспроизведите алгоритм быстрого поиска. Выполните трассировку алгоритма для нескольких различных массивов и искомым элементов.
22. Какая ошибка возможна при выполнении алгоритма быстрого поиска? Как блокировать появление этой ошибки?
23. Что представляет собой барьер в соответствующем алгоритме поиска? Куда и как выставляется барьер? Почему алгоритм с барьером эффективнее классического алгоритма поиска?
24. Почему алгоритмы линейного поиска считаются неэффективными? Какой алгоритм эффективнее линейного? Когда его можно применять?
25. Какой массив считается отсортированным? Как может быть отсортирован массив?
26. Опишите общую схему бинарного поиска. Как выбирается средний элемент массива? Как определяются новые границы области поиска?
27. В каком случае бинарный поиск завершается обнаружением искомого элемента? Что считается признаком отсутствия искомого элемента в массиве?
28. Воспроизведите алгоритм бинарного поиска. Выполните трассировку алгоритма для нескольких различных массивов и искомым элементов.
29. Какие критерии используются для сравнения алгоритмов между собой с точки зрения их эффективности?
30. Что называется входом алгоритма? Что представляет собой множество входов алгоритма?
31. Зачем понадобилось введение понятия «трудоемкость алгоритма»? Как связаны эффективность и трудоемкость алгоритма?
32. Как подсчитывается трудоемкость алгоритма? Что представляет собой функция трудоемкости?
33. Как определяется объемная эффективность? Что представляет собой функция объема памяти?
34. Что называется характерным размером входа? Как определяется характерный размер входа? Что называется длиной входа?

35. Почему при определении функций трудоемкости и объема памяти приходится обсуждать лучший и худший случаи. Как они определяются и подсчитываются?
36. Почему при анализе алгоритмов обычно интересуются худшим случаем?
37. Сравните между собой функции трудоемкости алгоритмов поиска классического, быстрого и с барьером.
38. Как корректно выбрать лучший и худший случаи?
39. Дайте точное определение функций трудоемкости лучшего и худшего случаев.
40. Дайте точное определение функций объема памяти лучшего и худшего случаев.
41. Опишите схему определения функции трудоемкости бинарного поиска.
42. В чем проявляется преимущество алгоритма бинарного поиска перед алгоритмами линейного поиска?

Контрольные вопросы и упражнения к главе 11

1. Дайте определение понятий «кратный цикл», «вложенный цикл», «объемлющий цикл», «глубина вложения». Какой может быть глубина вложения?
2. На что следует обратить внимание в первую очередь во время анализа задачи, в которой выполняются действия над математическими структурами типа вектора, матрицы и т. д.?
3. Сформулируйте постановку задачи сложения матриц. Что считается суммой матриц?
4. Опишите рекомендуемую общую схему рассуждений в процессе построения кратного цикла.
5. Воспроизведите алгоритм 11.1 решения задачи нахождения суммы матриц, используя для построения циклов схему 0. Выполните трассировку алгоритма.
6. Опишите типичную ошибку, допускаемую при построении кратных циклов. К чему она приводит?
7. Как можно по математической формулировке задачи с векторами и матрицами построить цикл?
8. Как найти функции трудоемкости и объема памяти для алгоритма 11.1?
9. Распишите подробно выражения, определяющие нахождение суммы, произведения, суммы произведений, произведения сумм для элементов матрицы.
10. Воспроизведите алгоритм 11.2 решения задачи нахождения суммы произведений элементов строк матрицы, используя для построения циклов схему 0. Выполните трассировку алгоритма.
11. Постройте алгоритм нахождения суммы произведений элементов столбцов матрицы. Выполните трассировку алгоритма.
12. Постройте алгоритм нахождения произведения сумм элементов строк матрицы. Выполните трассировку алгоритма.
13. Постройте алгоритм нахождения произведения сумм элементов столбцов матрицы. Выполните трассировку алгоритма.
14. Дайте определение понятий «минмаксный», «максминный», «седловой» элемент матрицы.

15. Воспроизведите алгоритм 11.3 решения задачи нахождения минимального элемента матрицы, используя для построения циклов схему 0. Выполните трассировку алгоритма.
16. Постройте алгоритм решения задачи нахождения максимального элемента матрицы. Выполните трассировку алгоритма.
17. Постройте алгоритмы решения задачи нахождения максимального и минимального элементов матрицы. Выполните трассировку алгоритма.
18. В каких случаях можно упростить построение кратного цикла для типичных задач нахождения сумм, произведений, экстремальных элементов с элементами матриц?
19. Сформулируйте постановку задачи умножения матрицы на вектор. Какие варианты такого умножения допускаются?
20. Воспроизведите алгоритм 11.4 решения задачи умножения матрицы на вектор, используя для построения циклов схему 0. Выполните трассировку алгоритма. Определите функцию трудоемкости и объема памяти этого алгоритма.
21. Сформулируйте постановку задачи умножения матрицы на матрицу. Какие варианты такого умножения допускаются?
22. Воспроизведите алгоритм 11.5 решения задачи умножения матрицы на матрицу, используя для построения циклов схему 0. Выполните трассировку алгоритма. Определите функцию трудоемкости и объема памяти этого алгоритма.
23. Какой вид в общем случае имеет функция трудоемкости алгоритма с кратным циклом?
24. Какой вид в общем случае имеет функция объема памяти алгоритма, использующего векторы, матрицы и аналогичные математические объекты с более высокой размерностью?
25. В чем состоит важность задачи сортировки? Какие разновидности сортировки существуют?
26. Чем объясняется существование большого количества различных алгоритмов сортировки?
27. Опишите общую идею сортировки прямым выбором. Как связана эта идея с ранее изученными типичными алгоритмами?
28. Воспроизведите алгоритм 11.6 сортировки прямым выбором, используя для построения циклов схему 0.
29. Выполните трассировку построенного алгоритма прямым выбором. Какие недостатки имеет трассировка как метод тестирования алгоритмов?
30. Определите функцию трудоемкости и объема памяти алгоритма сортировки прямым выбором.
31. Опишите общую идею сортировки прямыми вставками. Как связана эта идея с ранее изученными типичными алгоритмами?
32. Воспроизведите алгоритм 11.7 сортировки прямыми вставками, используя для построения циклов схему 0.
33. Выполните трассировку построенного алгоритма прямыми вставками.
34. Определите функцию трудоемкости и объема памяти алгоритма сортировки прямыми вставками.

35. Опишите общую идею обменной сортировки.
36. Воспроизведите алгоритм 11.8 обменной сортировки, используя для построения циклов схему 0.
37. Выполните трассировку построенного алгоритма обменной сортировки.
38. Определите функцию трудоемкости и объема памяти алгоритма обменной сортировки.
39. Сравните между собой базовые алгоритмы сортировки по эффективности.

Контрольные вопросы к главе 12

1. Перечислите основные отличительные черты компьютера как устройства, используемого для решения различных задач.
2. Дайте определения понятий «компьютер», «архитектура компьютера», «интерфейс», «аппаратное обеспечение».
3. Охарактеризуйте основные группы устройств, входящих в состав компьютера.
4. Какую структуру имеет память компьютера? Что обеспечивает такая структура?
5. Дайте определения терминов «бит», «байт», «поле», «длина поля».
6. Какими функциональными свойствами обладает бит с точки зрения необходимости хранения данных и программ?
7. В каких смыслах может использоваться термин «бит»?
8. Чем с организационной точки зрения бит отличается от байта?
9. Зачем понадобилось объединять биты в байты и поля? Как нумеруются биты в пределах байта и поля?
10. Что считается объемом памяти? В каких единицах она измеряется?
11. Дайте определения основных кратных единиц объема.
12. Какие виды памяти используются в персональном компьютере?
13. Для чего нужна оперативная память? Охарактеризуйте ее отличительные особенности.
14. Что может находиться в оперативной памяти компьютера?
15. Что представляет собой адрес байта или поля памяти? Почему оперативная память считается прямоадресуемой?
16. Что называется адресным пространством компьютера? Чем адресное пространство отличается от объема оперативной памяти?
17. Для чего нужен кэш в составе компьютера? С какими устройствами взаимодействует кэш? Каким может быть объем кэша?
18. Какую роль играет постоянная память? Зачем нужен BIOS?
19. Для чего нужна внешняя память? Охарактеризуйте ее отличительные особенности.
20. Сравните между собой оперативную и внешнюю память.
21. Дайте определения понятий «загрузка программы» и «пуск программы».
22. Какие две группы внешней памяти принято выделять?
23. Охарактеризуйте накопители на жестких магнитных дисках.

24. Дайте определения понятий «рабочая поверхность», «дорожка», «сектор», «кластер».
25. Как определить объем магнитного дискового носителя информации?
26. Для чего необходимо выполнять форматирование магнитных дисков?
27. Какие проблемы сопровождают хранение программ и данных на жестких дисках?
28. Как происходит фрагментация жесткого диска? Чем она вредна?
29. Охарактеризуйте твердотельные накопители. В чем преимущество твердотельных накопителей по сравнению с жесткими магнитными дисками?
30. Охарактеризуйте накопители на оптических дисках.
31. Сравните между собой жесткие магнитные и оптические диски.
32. Охарактеризуйте устройства флеш-памяти.
33. Сколько может быть встроенных дисковых устройств в персональных компьютерах? Как они обозначаются?
34. Что называется процессором? Опишите его основные функции.
35. Дайте определения понятий «система команд», «машинная команда», «машинная программа», «регистр», «сумматор».
36. Опишите схему выполнения машинной программы.
37. Перечислите основные технические характеристики процессоров.
38. Как связаны между собой быстродействие и тактовая частота процессора?
39. Что понимается под машинным словом? Как влияет длина машинного слова на производительность компьютера?
40. Для чего нужен тактовый генератор?
41. Для чего нужна шина? Из каких элементов она состоит?
42. Какие технические характеристики используются для описания шины?
43. Что определяется разрядностью и пропускной способностью шины?
44. Что такое порт компьютера?
45. Сколько шин может быть в компьютере?
46. Нарисуйте упрощенную схему устройства персонального компьютера.
47. Какие платы используются в персональных компьютерах?
48. Для чего необходима материнская плата?
49. Для чего нужен системный блок?
50. Какие устройства компьютера находятся в системном блоке?
51. Что находится на лицевой панели системного блока?
52. Что находится на задней панели системного блока?
53. Охарактеризуйте назначение устройств ввода-вывода.
54. Что называется дисплеем? Дайте классификацию дисплеев.
55. Какие технические характеристики используются для описания дисплеев?
56. Как классифицируются дисплеи по принципу действия?
57. Назовите типовые размеры экранов дисплеев, используемых в настоящее время.
58. Что определяет разрешающая способность? Какова разрешающая способность современных дисплеев?

59. Что определяет класс защиты дисплея?
60. Для чего нужны адаптеры и контроллеры?
61. Назовите основные режимы работы клавиатуры.
62. Для чего нужны функциональные клавиши?
63. Что называется сочетанием клавиш?
64. Что называется текстовым курсором?
65. Объясните, как происходит прокрутка текста.
66. Что называется экранной страницей?
67. Опишите основные способы перемещения текстового курсора.
68. Для чего нужна компьютерная мышь?
69. Укажите основные технические характеристики и разновидности принтеров.
70. Сравните между собой принтеры различных принципов действия.
71. Что определяет разрешение при печати? Какое разрешение характерно для современных принтеров?
72. Для чего нужен сканер? Укажите технические характеристики сканеров.
73. Какие устройства должны входить в состав компьютера, чтобы он мог работать в мультимедийной среде?
74. Какие устройства относятся к аппаратному обеспечению компьютерных сетей?
75. Что называется линией связи? Что может использоваться в качестве линий связи?
76. Перечислите основные характеристики линий связи.
77. Какие бывают сети? По каким признакам сеть относят к той или иной группе?
78. Охарактеризуйте локальные сети.
79. Чем отличаются друг от друга однородные и неоднородные сети?
80. Что называется сегментом сети? Для чего нужны повторители и репитеры?
81. Дайте характеристику городским сетям.
82. Чем отличаются шлюзы от мостов?
83. Какое устройство необходимо для подсоединения компьютера к информационной сети через телефонную сеть? Какую роль играет это устройство?
84. Дайте характеристику региональным и глобальным сетям.
85. Что называется пропускной способностью сети? Назовите типовые значения пропускной способности современных сетей.
86. Что относится к ресурсам компьютера?
87. Какие ресурсы считаются сетевыми, а какие — локальными?
88. Чем отличается одноранговая сеть?
89. Охарактеризуйте сеть типа «клиент–сервер».
90. Что может использоваться в качестве терминала?
91. Что называется сервером? Какие бывают серверы?
92. Какие компьютеры считаются программно-совместимыми?
93. Что называется семейством компьютеров? Как понимать термин «аппаратная платформа»?

94. Сформулируйте правила безопасной для здоровья работы на компьютере.
95. Сформулируйте правила безопасной для аппаратуры работы на компьютере.
96. Сформулируйте правила эксплуатации мониторов.
97. Сформулируйте правила обращения с оптическими дисками.
98. Сформулируйте правила работы с мышью.

Контрольные вопросы к главе 13

1. Какую роль играют программы в работе компьютера?
2. Дайте определение понятия «программное обеспечение».
3. Каким образом можно узнать назначение, основные возможности и способы управления программами?
4. Что понимается под терминами software и hardware?
5. Что относится к ресурсам компьютера?
6. Сравните между собой содержание понятий «аппаратное обеспечение», «программное обеспечение» и «ресурсы компьютера».
7. Дайте определение понятий «пакет программ», «программная система», «программный продукт».
8. Опишите классификацию программного обеспечения.
9. Для чего используется системное программное обеспечение? Назовите основные программы, относящиеся к системному программному обеспечению.
10. Зачем нужны операционные системы?
11. Для чего необходимы программы-оболочки?
12. Для чего служат утилиты?
13. Какую роль играют драйверы?
14. Для чего используется инструментальное программное обеспечение?
15. Дайте определение понятий «прикладная программа» и «приложение».
16. Для чего используются текстовые и графические редакторы. Приведите примеры редакторов.
17. Для чего используются электронные таблицы?
18. Что называется базой данных? Приведите примеры баз данных.
19. Для чего нужны СУБД?
20. В чем заключается основная особенность интегрированных систем?
21. Опишите известные вам пакеты прикладных программ.
22. Что называется файловой системой диска и из чего она состоит?
23. Что называется каталогом и какую информацию он содержит?
24. Какую структуру имеет каталог?

25. Дайте определение понятий «корневой каталог», «подкаталог», «подкаталог первого уровня», «подкаталог второго уровня», «родительский каталог», «дочерний подкаталог», «корневая папка», «папка».
26. Что называется маршрутом к файлу и почему его необходимо указывать?
27. Что называется спецификацией файла?
28. Чем отличается модификация программы от ее версии?
29. Как обозначаются новые версии и модификации?
30. Что понимается под альфа- и бета-версиями программы?
31. Какие версии операционной системы считаются локализованными?
32. Каким образом распространяются пакеты программ?
33. Что называется дистрибутивом и для чего необходима инсталляция?
34. Что называется операционной системой?
35. Охарактеризуйте основные функции операционных систем.
36. Какой диск называется системным? Какие диски могут быть системными?
37. Какие бывают операционные системы? Охарактеризуйте с этой точки зрения семейства Windows 9x и Windows NT/2000/XP.
38. Что понимается под программной платформой?
39. Что называется интерфейсом пользователя?
40. Опишите основные особенности текстового интерфейса.
41. Что называется оболочкой? Какие оболочки вам известны?
42. Сравните между собой текстовый и табличный интерфейсы.
43. Опишите основные особенности табличного и графического интерфейсов и сравните их между собой.
44. Что называется файлом? Что может находиться в файле?
45. Какие операции могут выполняться над файлами?
46. Охарактеризуйте операции создания и уничтожения файла.
47. Охарактеризуйте операции открытия и закрытия файла.
48. Сравните между собой операции создания и открытия, а также операции уничтожения и закрытия файла.
49. Сравните между собой операции копирования и перемещения. Можно ли обойтись без отдельной операции перемещения?
50. Охарактеризуйте операции редактирования и переименования файла.
51. Что называется атрибутом? Какие атрибуты файла являются основными?
52. Как задается название файла в операционной системе Windows?
53. Какую роль играет расширение и как оно задается? Что называется полным именем?
54. Дайте определение понятий «программный файл», «командный файл», «выполняемый файл», «резервный файл», «файл помощи», «драйвер». Укажите соответствующие расширения.
55. Для чего необходимо групповое имя? Как оно образуется?

Контрольные вопросы к главе 14

1. Какие разновидности компьютерных данных считаются базовыми? Как с ними связаны остальные виды данных?
2. Сравните между собой принципы равномерного и неравномерного кодирования. Какую возможность предоставляет использование неравномерного кодирования.
3. Охарактеризуйте основные области применения равномерного и неравномерного кодирования.
4. В чем проявляется основная проблема неравномерного кодирования? Какими способами она может быть решена?
5. Почему на практике не применяется кодирование с ограничителями?
6. Какие коды называются префиксными? Сформулируйте условие Фано.
7. Какие коды называются статистическими?
8. Опишите способ построения префиксных кодов, предложенный Шенноном и Фано. Сравните избыточность этого способа и равномерного кодирования.
9. Что происходит во время прямого прохода построения кодовой таблицы Хаффмана?
10. Как осуществляется обратный проход построения кодовой таблицы Хаффмана?
11. Дайте определение понятий «вершина графа», «ребро графа», «граф». Приведите примеры этих понятий.
12. Какой граф считается взвешенным? Приведите примеры весов вершин и ребер графа.
13. Что называется путем в графе? Что считается длиной пути в графе?
14. Что такое дерево? Чем отличается дерево от общего случая графа?
15. Дайте определение понятий «корень дерева», «родительский узел», «дочерний узел», «лист».
16. Сколько дочерних и родительских узлов может иметь обычный узел дерева и корень дерева?
17. Опишите способ построения кодового дерева Хаффмана.
18. Опишите порядок кодирования с помощью кодового дерева Хаффмана.
19. Опишите порядок декодирования с помощью кодового дерева Хаффмана.
20. Как по кодовому дереву можно определить среднюю длину кодовой цепочки и относительную избыточность построенного кода Хаффмана.
21. Чем знаменит код Хаффмана? Где он применяется кроме неравномерного кодирования текстов?
22. Сформулируйте принцип группового кодирования.
23. Опишите общую схему адаптивного кодирования.
24. Что представляет собой блочное кодирование? В чем преимущество блочного кодирования по сравнению с алфавитным?
25. К какой группе кодов относится арифметическое кодирование?
26. Опишите схему построения арифметического кода сообщения.
27. Опишите схему декодирования арифметического кода.
28. Какие проблемы возникают во время построения арифметического кода? Как их можно решить?

29. Сформулируйте основную идею словарного кодирования.
30. Дайте определения понятий «скользящее окно», «словарь», «буфер». Как выбираются их длины?
31. Как формируется ссылка при словарном способе кодирования? Что представляет собой код в этом способе?
32. Опишите общую схему получения кода в словарном способе.
33. Опишите схему декодирования в словарном способе.
34. Что можно сказать об относительной избыточности словарного способа кодирования? Какое сжатие сообщения достигается при использовании этого способа?
35. Дайте определения понятий «анимация», «видео», «мультимедийная технология».
36. Как в компьютерах кодируется графическая информация?
37. Дайте определения понятий «пиксел», «растр», «разрешающая способность», «сканирование».
38. Опишите принципы кодирования монохромных изображений.
39. Дайте определения понятий «цветовая система», «цветовая координата», «цветовой вектор», «цветовое пространство».
40. Опишите принципы кодирования цветных изображений с использованием цветовой схемы RGB.
41. Опишите особенности полноцветного способа, режима High Color и индексного режима представления цветной графики.
42. Опишите особенности и области применения субтрактивной цветовой схемы.
43. Охарактеризуйте области использования цветковых систем YCbCr, XYZ и Lab.
44. Почему принципы сжатия текстовых данных не подходят для графических данных?
45. Опишите общие подходы к сжатию графических данных.
46. В чем преимущество и в чем основной недостаток растровой графики?
47. Как осуществляется кодирование изображения при использовании векторной графики?
48. Какие возможности для работы с изображениями предоставляет векторная графика?
49. Что такое графический формат? Охарактеризуйте основные графические форматы.
50. Как осуществляется преобразование звука в цифровую форму?
51. Какие параметры характеризуют цифровую форму звука?
52. Охарактеризуйте звуковой формат WAV.
53. Как подсчитывается битрейт в выбранном способе оцифровывания звука?
54. Для чего используется разностное кодирование? Как оно осуществляется?
55. Какие физические факторы используются для сжимающего кодирования звука?
56. Охарактеризуйте звуковой формат MP3.
57. Какие сложности возникают при кодировании видеоданных?
58. Какие способы сжатия используются при кодировании видеоданных?
59. Охарактеризуйте формат кодирования MPEG.
60. Охарактеризуйте формат кодирования HDTV.

61. Что называется кодом Грея?
62. В чем отличие одношаговых кодов от многошаговых. Приведите примеры многошаговых кодов.
63. Дайте определение понятий «вес Хемминга», «расстояние Хемминга». Как рассчитываются эти величины?
64. Опишите простейший способ построения кода Грея. На каком свойстве этого кода он основан?
65. Как по прямому двоичному коду получить код Грея?
66. Как по коду Грея восстановить прямой двоичный код?
67. Опишите основные технические области применения кодов Грея.
68. Опишите способы применения кодов Грея в компьютерных областях.
69. Что называется топологией компьютерной системы?
70. Охарактеризуйте типичные топологии вычислительных систем.
71. В чем заключается основной недостаток полносвязной топологии?
72. Сравните между собой полносвязную, кольцевую и матричную топологию.
73. Что такое гиперкуб? Как осуществляется эффективная нумерация вершин гиперкуба?
74. В чем преимущество топологии гиперкуба перед другими топологиями?
75. Сформулируйте принцип обезличенности кода.

Контрольные вопросы к главе 15

1. Охарактеризуйте возможные причины появления ошибок в кодах данных при их хранении и передаче в компьютерах.
2. Опишите классическую схему передачи сообщений по линиям связи.
3. Чем отличается линия связи от канала связи?
4. Каким образом передаются сообщения в компьютерных линиях связи?
5. Чем реальный импульс отличается от идеального?
6. Сколько различных уровней должен иметь импульс, служащий для передачи сообщений в компьютерных линиях связи?
7. Почему в линиях связи передаются не все частоты? Что называется полосой пропускания линии связи? Как определяется ширина полосы пропускания?
8. Что называется пропускной способностью канала? В каких единицах принято измерять пропускную способность канала?
9. Что ограничивает барьер Найквиста?
10. Чем отличается скорость передачи сообщения от пропускной способности канала? Как они связаны?
11. Чем реальные каналы связи отличаются от идеальных?
12. На что влияет наличие помех в реальных каналах связи?
13. Как определяется пропускная способность реального канала связи?
14. Как можно увеличить пропускную способность канала связи?

15. Какая передача сообщения считается надежной?
16. Сформулируйте теорему Шеннона о кодировании при наличии помех.
17. Чем обеспечивается надежность передачи сообщений при наличии помех?
18. Как рассчитывается относительная избыточность сообщения?
19. Чем отличается кодирование источника от кодирования канала связи?
20. Опишите схему передачи сообщений по линиям связи с отдельным кодированием источника и канала.
21. Как можно обнаружить единичную ошибку в байте? Почему этим способом невозможно исправить ошибку?
22. Опишите общую схему исправления ошибки, предложенную Хеммингом. За счет чего создается возможность исправить ошибку в этой схеме?
23. Из каких частей состоит кодовое слово?
24. Чем отличается систематический код от несистематического?
25. Какой код называется линейным?
26. Дайте определения понятий «порождающая матрица», «вектор ошибок», «проверочная матрица», «синдром», «кодовое слово», «принятое слово».
27. Опишите общую схему кодирования и декодирования линейных блочных кодов.
28. Какие вектора участвуют в процедурах кодирования и декодирования? Укажите их длины.
29. Опишите структуру порождающей матрицы. Как она строится?
30. Как с помощью порождающей матрицы осуществляется кодирование?
31. Какую структуру имеет проверочная матрица? Как она строится?
32. Как с помощью проверочной матрицы осуществляется декодирование?
33. Как определяется наличие ошибки в принятом коде?
34. Как определяется местоположение ошибки в принятом коде? Как исправляется найденная ошибка?
35. Что произойдет, если осуществить стандартный процесс декодирования, но ошибка в принятом коде окажется кратной?
36. Опишите идею метода построения несистематического кода Хемминга? Какие ошибки могут быть исправлены этим способом? В чем преимущество этого способа кодирования перед стандартным способом блочного кодирования?
37. Опишите общую схему построения несистематического кода Хемминга.
38. Как вставляются контрольные разряды в исходное слово при построении несистематического кода Хемминга?
39. Как при построении несистематического кода Хемминга формируются группы информационных битов, связанные с каждым из контрольных?
40. Как формируется значение контрольных разрядов при построении несистематического кода Хемминга?
41. Как определяется положение ошибочного бита в несистематическом коде Хемминга?
42. Где используются линейные блочные коды Хемминга?

43. Что называется корректирующей способностью помехоустойчивого кода? Какой она может быть?
44. Что называется расстоянием Хемминга между двумя двоичными кодами?
45. Что называется минимальным расстоянием Хемминга линейного кода? Как определяется это расстояние на практике?
46. Каким должно быть минимальное расстояние Хемминга у помехоустойчивого кода?
47. Какое обстоятельство обеспечивает возможность обнаружения и исправления ошибок, попавших в хранящийся или передаваемый код?
48. Как влияет значение минимального расстояния Хемминга на корректирующую способность кода?
49. Что называется сферой кодового слова?
50. Опишите схему построения порождающей матрицы линейного (n, k) -кода с заданным минимальным расстоянием Хемминга.
51. Какие коды считаются совершенными? Приведите примеры совершенных кодов.
52. Как определить количество контрольных разрядов, которые необходимо включить в кодируемый блок, чтобы обеспечить корректирующую способность в t ошибок?
53. Какое неравенство связывает между собой все три параметра линейного блочного кода? Как это неравенство ограничивает минимальное расстояние Хемминга этого кода?
54. Что определяет граница Хемминга помехоустойчивого кода?
55. Опишите общую схему построения линейного блочного (n, k, d_{\min}) -кода.
56. В чем проявляется основной недостаток линейных блочных кодов?
57. Дайте определение операции целочисленного деления.
58. Какие числа считаются сравнимыми по модулю? Что называется модулем сравнения?
59. Чем отличаются обычные арифметические операции от этих же операций, но выполняемых по модулю 2?
60. Как выполняются сложение и умножение по модулю 2 для многоразрядных чисел?
61. Что называется алгебраическим полем? Приведите примеры полей и числовых множеств, не являющихся полями.
62. Какое поле называется полем Галуа? Какими параметрами характеризуются поля Галуа?
63. Что называется расширением поля? Что такое подполе?
64. Дайте определения понятий «порядок полинома», «корень полинома», «минимальный полином», «нулевой полином».
65. Сколько корней имеет полином? Где могут находиться эти корни?
66. Как осуществляется сложение и вычитание полиномов. Напишите соответствующие математические соотношения.
67. Как осуществляется умножение полиномов? Напишите соответствующие математические соотношения.
68. Какие полиномы считаются приводимыми, неприводимыми и простыми?
69. Опишите способы задания полиномов, применяющиеся в помехоустойчивом кодировании.

70. Сформулируйте правило деления полиномов. Какие делители полинома считаются тривиальными?
71. Как определяется наименьшее общее кратное полиномов?
72. Как определяется наибольший общий делитель полиномов?
73. Какие полиномы считаются сравнимыми по модулю?
74. Как осуществляются арифметические операции по модулю над полиномами?
75. В чем проявляется основная особенность операций над полиномами, заданными над полем Галуа?
76. Как представить двоичный код в виде полинома над полем Галуа?
77. Сравните между собой основные способы представления двоичных кодов, применяемые в помехоустойчивом кодировании.
78. Что называется циклическим сдвигом? Какие сдвиги имеют смысл?
79. Как задаются сдвиги при использовании полиномиального представления двоичных кодов?
80. Какой линейный код считается циклическим?
81. Дайте определение понятий «порождающий полином», «кодировый полином», «полином ошибки», «принятый полином», «проверочный полином», «синдромный полином».
82. Как связаны между собой кодировый и порождающий полиномы?
83. Как строятся порождающие полиномы циклического (n, k) -кода? Как на практике выбираются порождающие полиномы?
84. Как осуществляется кодирование с применением порождающего полинома?
85. Как осуществляется декодирование циклических кодов?
86. Что такое пакет ошибок?
87. Для чего используются CRC-коды? Что представляет собой контрольная сумма? Как определяется наличие ошибки при использовании CRC-кодов? Как в этом случае исправляется ошибка?
88. Для чего в помехоустойчивом кодировании используются поля Галуа $GF(2^{\bar{m}})$?
89. Как выполняются сложение и вычитание над элементами полей Галуа $GF(2^{\bar{m}})$?
90. В чем проявляется основная проблема при задании полей Галуа $GF(p^{\bar{m}})$ для $\bar{m} > 1$?
91. Дайте определение понятий «примитивный элемент» и «примитивный полином».
92. Как на практике находят примитивные полиномы?
93. Как задается операция умножения в поле Галуа с помощью примитивного элемента?
94. Как выполняются умножение и деление в поле Галуа с применением таблиц логарифмов и антилогарифмов?
95. Как осуществляются действия над полиномами, заданными над полем $GF(2^{\bar{m}})$?
96. В чем проявляется основная особенность БЧХ-кодов?
97. Каким свойством обладает порождающий полином, построенный по корням, которые принадлежат множеству последовательных степеней примитивного элемента поля Галуа?
98. Какие элементы поля Галуа считаются сопряженными?

99. Как построить минимальный полином с помощью сопряженных элементов поля Гауа?
100. Как строится порождающий полином БЧХ-кода?
101. Опишите общую схему построения БЧХ-кода.
102. Почему общую схему декодирования циклических кодов нельзя применить для БЧХ-кодов?
103. В чем состоит прямое решение Питерсона—Горенштейна—Цирлера?
104. Дайте определение понятий «синдром», «локатор ошибки».
105. Как строится полином локаторов ошибки?
106. Как строится система линейных уравнений, служащих для определения локаторов ошибок?
107. Как определяется фактическое количество ошибок в принятом слове в БЧХ-кодах?
108. Для чего применяется процедура Ченя? В чем она состоит?
109. Опишите общую схему декодирования БЧХ-кода.
110. В чем отличие кодов Рида—Соломона от обычных БЧХ-кодов?
111. Какими корректирующими свойствами обладают коды Рида—Соломона?
112. Где нашли применение коды Рида—Соломона?

Контрольные вопросы к главе 16

1. Опишите общую идею применения подпрограмм (вспомогательных алгоритмов).
2. В чем проявляются преимущества использования подпрограмм при построении алгоритмов?
3. Дайте определения понятий «вспомогательный алгоритм», «подпрограмма», «вызывающий алгоритм», «точка вызова», «вызов подпрограммы».
4. Какие задачи приходится решать, используя прямое включение вспомогательного алгоритма в основной? В чем проявляются недостатки этого способа?
5. Что называется описанием подпрограммы? Какие требования предъявляются к этой конструкции?
6. Из каких элементов состоит заголовок подпрограммы? Что называется формальными параметрами? Какие разновидности формальных параметров встречаются в описании подпрограммы?
7. Какую структуру имеет вызов подпрограммы? Где размещается вызов подпрограммы?
8. Что называется фактическими параметрами? Какими могут быть фактические параметры?
9. Сравните между собой фактические и формальные параметры. Каким должно быть соответствие между ними?
10. Чем отличается описание подпрограммы-процедуры от описания подпрограммы-функции?

11. В чем основное отличие вызова подпрограммы-процедуры от вызова подпрограммы-функции?
12. Приведите примеры проявления самоподобия в различных областях.
13. Что называется рекурсией? Для чего она используется?
14. Сравните между собой итерационные и рекурсивные алгоритмы.
15. На чем основано выполнение рекурсивных подпрограмм?
16. Какая ветвь алгоритма считается терминальной?
17. Как осуществляется прямой ход выполнения рекурсивной подпрограммы?
18. Опишите действия, выполняемые во время обратного хода.
19. Каковы недостатки применения рекурсивных подпрограмм?
20. В чем достоинства применения рекурсивных подпрограмм?
21. Когда следует и когда не следует применять рекурсивные подпрограммы?
22. Дайте определения понятий «параллельная рекурсия», «дерево рекурсии», «уровень рекурсии», «глубина рекурсии».
23. Что называется полным обходом бинарного дерева? Какими могут быть полные обходы?
24. Какие алгоритмы имеют экспоненциальную трудоемкость?
25. Сформулируете общие принципы построения рекурсивных подпрограмм.
26. Сколько терминальных ветвей должна иметь рекурсивная подпрограмма, возвращающая логическое значение?
27. На каких представлениях основано определение алгоритма в теории Геделя и Черча?
28. Что считается длиной входа алгоритма?
29. Какие функции считаются вычислимыми, а какие невычислимыми?
30. Какие функции называются частичными, а какие всюду определенными?
31. Охарактеризуйте разновидности рекурсивных функций. Как они связаны друг с другом?
32. Какие функции считаются примитивно рекурсивными? Являются ли примитивно рекурсивные функции частично определенными?
33. Дайте определения функций нуля, следования и проекции.
34. Дайте определения операторов суперпозиции и примитивной рекурсии.
35. Для чего используется оператор примитивной рекурсии? Почему он так называется?
36. Дайте определения операций сложения и умножения с помощью примитивно рекурсивных функций.
37. Для чего понадобилось введение оператора минимизации?
38. Сформулируйте определение оператора минимизации.
39. Как с помощью оператора минимизации ввести операцию вычитания?
40. Почему функции, построенные с использованием оператора минимизации, считаются частично определенными?
41. Какая функция считается общерекурсивной?
42. Какие задачи считаются алгоритмически неразрешимыми?

43. Сформулируйте тезис Черча. В чем его теоретическое значение?
44. Опишите общую схему доказательства алгоритмической разрешимости или неразрешимости проблемы.

Контрольные вопросы к главе 17

1. Что называется автоматом? Опишите общий способ использования различных автоматов человеком.
2. Чем аналоговые автоматы отличаются от дискретных?
3. Что представляет собой абстрактный автомат? Что входит в состав абстрактного автомата? Нарисуйте схему абстрактного автомата.
4. Что называется состоянием автомата?
5. Что определяют функции выхода и перехода? Дайте их строгое математическое определение.
6. Дайте математически строгое определение абстрактного автомата.
7. Какую роль играют абстрактные автоматы в теории автоматов?
8. Охарактеризуйте основные уточнения, конкретизации модели абстрактного автомата.
9. Опишите классификацию по способу просмотра входного сообщения.
10. Опишите классификацию автоматов по наличию памяти.
11. Опишите классификацию по виду функции переходов.
12. Чем отличаются автоматы Мили от автоматов Мура?
13. Чем отличается подход Тьюринга от подхода частично-рекурсивных функций в определении понятия «алгоритм»?
14. Дайте определение машины Тьюринга с точки зрения классификации абстрактных автоматов.
15. Как понимается неограниченность ленты машины Тьюринга? Что может находиться на ленте?
16. Опишите возможности головки машины Тьюринга.
17. Опишите общее устройство машины Тьюринга. Нарисуйте ее схему.
18. Что моделировал Тьюринг, создавая свою модель?
19. Дайте определение понятий «внешний алфавит», «внутренний алфавит», «начальное состояние», «завершающее состояние», «конфигурация», «функция переходов» применительно к машине Тьюринга.
20. В чем проявляются особенности функции переходов машины Тьюринга?
21. Опишите действия машины Тьюринга на одном такте.
22. Дайте формальное определение машины Тьюринга.
23. Опишите итерационный процесс работы машины Тьюринга.
24. Что является результатом работы машины Тьюринга?
25. В каких случаях входная цепочка принимается машиной Тьюринга, а в каких — не принимается?
26. Какую структуру имеет команда машины Тьюринга?

27. С какой управляющей структурой сопоставима команда машины Тьюринга?
28. Что отличает программу машины Тьюринга от программы машины Поста и обычных компьютерных программ?
29. Какими способами можно представить программу машины Тьюринга?
30. Что значит построить машину Тьюринга?
31. Воспроизведите ход рассуждений при построении программы машины Тьюринга для вычисления значения функции следования.
32. Воспроизведите программу машины Тьюринга для вычисления значения функции следования в табличной форме.
33. Воспроизведите программу машины Тьюринга для вычисления значения функции следования в форме графа.
34. Что представляет собой протокол работы машины Тьюринга? Как он строится?
35. Воспроизведите программу машины Тьюринга для вычисления значения базовой функции нуля.
36. Воспроизведите программу машины Тьюринга для вычисления значения базовой функции проекции.
37. Постройте машину Тьюринга для вычисления значений функции $y = 2x + 1$.
38. Постройте машину Тьюринга для вычисления значений функции $\text{trunc}(x/2)$.
39. Постройте машину Тьюринга для вычисления значений функции $\text{odd}(x)$.
40. Найдите результат выполнения приведенной в таблице программы, если на ленте число x представлено $x + 1$ единицей

	q_1	q_2	q_3	q_4	q_5
0	$q_2 \lambda R$	$q_3 \lambda R$	$q_3 1 R$	—	—
1	$q_1 \lambda S$	$q_4 1 S$	$q_4 1 R$	$q_5 1 R$	—

41. Найдите результат выполнения приведенной в таблице программы, если на ленте число x представлено $x + 1$ единицей

	q_1	q_2	q_3	q_4	q_5	q_6
0	$q_2 \lambda R$	$q_6 \lambda R$	—	$q_5 1 S$	$q_4 \lambda R$	$q_3 1 S$
1	$q_1 \lambda S$	$q_3 1 S$	—	$q_3 1 S$	$q_3 1 S$	$q_3 1 S$

42. Что называется композицией машин Тьюринга? Опишите основные способы создания композиций машин Тьюринга.
43. Перечислите основные разновидности машин Тьюринга.
44. Охарактеризуйте отличительные особенности недетерминированных машин Тьюринга.
45. Каким образом могут быть реализованы недетерминированные машины Тьюринга?
46. Как определяются результаты работы недетерминированной машины Тьюринга?

47. Чем отличается вероятностная машина Тьюринга от классической и недетерминированной?
48. Перечислите и охарактеризуйте структуры машин Тьюринга с разными вариантами лент.
49. Что такое эмуляция? Для чего она требуется?
50. Чем отличается универсальная машина Тьюринга? На чем основана возможность построения такой машины?
51. Какие исполнители и системы считаются полными по Тьюрингу? Приведите примеры.
52. Опишите компьютерную архитектуру фон Неймана, нарисуйте принципиальную схему компьютера с такой архитектурой.
53. Проследите связь между различными вариантами структур машин Тьюринга и классификацией компьютерных архитектур.
54. Какие архитектуры компьютеров считаются параллельными? Приведите примеры.
55. Чем отличается тезис от других математических утверждений?
56. Сформулируйте тезис Тьюринга и сравните его с тезисом Черча.
57. В чем проявляется теоретическое и практическое значение тезиса Тьюринга—Черча?
58. Какая проблема считается массовой? Приведите примеры массовых проблем.
59. Какие проблемы считаются алгоритмически неразрешимыми? Какую роль играет доказательство алгоритмической неразрешимости проблемы? Приведите примеры неразрешимых проблем.
60. Что называется формальной теорией? Какие теории считаются полными, а какие неполными? Приведите примеры.
61. Сформулируйте проблему остановки машины Тьюринга. Как она решена?
62. Какую роль играет доказательство неразрешимости проблемы остановки?
63. Как связаны теоретические проблемы остановки и эквивалентности с практикой программирования?
64. Дайте определение магазинного автомата с точки зрения классификации абстрактных автоматов. Почему они так называются? Для чего используется эта модель?
65. Как может быть устроена внешняя память магазинного автомата. Что такое стек?
66. Опишите общее устройство магазинного автомата. Нарисуйте его схему.
67. Сравните устройство машины Тьюринга и машинного автомата. В чем они похожи и чем различаются?
68. Что считается конфигурацией магазинного автомата? Дайте формальное определение магазинного автомата.
69. Опишите работу магазинного автомата. Когда завершается его работа?
70. Что считается результатом работы магазинного автомата? Что называется языком магазинного автомата?
71. В чем проявляются недостатки машины Тьюринга? Какие модели устраняют эти недостатки? Какими свойствами обладают эти модели?
72. Как устроена память модели РАМ? Какими свойствами обладает регистр модели РАМ?

73. Каким устройствам обычного компьютера соответствует память РАМ?
74. Охарактеризуйте модель РАМ с точки зрения ее использования.
75. Опишите общее устройство РАМ. Нарисуйте ее схему.
76. Что считается состоянием РАМ? Почему РАМ не описываются функциями перехода?
77. Какую роль играет счетчик адреса в выполнении программ РАМ?
78. Как хранятся программы и обрабатываемые данные в РАМ?
79. Опишите общую структуру команды РАМ.
80. Что такое операнд? Какими могут быть операнды в командах РАМ? Как определяется значение операнда?
81. Охарактеризуйте группы команд, входящих в систему команд РАМ.
82. Приведите примеры действий, выполняемых командой загрузки РАМ, и опишите их.
83. Приведите примеры действий, выполняемых арифметическими командами РАМ, и опишите их.
84. Для чего нужны команды перехода? Какие команды перехода существуют в системе команд РАМ?
85. Опишите общую схему выполнения программы РАМ. Какое завершение работы программы РАМ считается корректным?
86. Когда входная цепочка РАМ считается допустимой? Что называется языком РАМ?
87. Сравните между собой РАМ, машину Тьюринга и магазинный автомат как различные варианты абстрактных автоматов.
88. Как определяются функции трудоемкости и объема памяти для модели РАМ? Какие критерии для этого используются?
89. Опишите основные этапы разработки программы для машины РАМ.
90. Для чего выполняется распределение регистров? Как это делается?
91. Как организуется ввод исходных данных в программах РАМ?
92. Воспроизведите программу РАМ1 и оцените ее сложность.
93. Как выполняются безусловные переходы при выполнении программ РАМ?
94. Как выполняются условные переходы при выполнении программ РАМ?
95. Опишите типичную схему организации ветвления в программах РАМ.
96. Воспроизведите программу РАМ2 и оцените ее сложность.
97. Опишите типичную схему организации цикла в программах РАМ.
98. Воспроизведите программу РАМ3 и оцените ее сложность.
99. Дайте определение конечного автомата с точки зрения классификации абстрактных автоматов.
100. Опишите общее устройство конечного автомата. Нарисуйте его схему.
101. Сравните устройство машины Тьюринга, машинного автомата и конечного автомата. В чем они похожи и чем различаются?
102. Что считается конфигурацией конечного автомата? Дайте формальное определение.
103. Опишите работу конечного автомата. Когда и чем завершается его работа?

104. Что считается языком конечного автомата? Какие конечные автоматы считаются эквивалентными?
105. Опишите основные области применения модели конечного автомата.
106. Опишите конечный автомат, который может рассматриваться как модель элементарного устройства памяти — бита. Задайте его функцию перехода в табличном виде и в виде графа.
107. Опишите общие принципы программной реализации конечного автомата.
108. Для чего нужны преобразования конечного автомата. Опишите схему стандартно выполняемых преобразований конечных автоматов. Какие требования предъявляются к таким преобразованиям?
109. Какой автомат считается частично определенным? В чем его недостатки?
110. Каким образом осуществляется переход к полностью определенному конечному автомату? Как соотносятся между собой исходный частично определенный и построенный на его базе полностью определенный конечный автомат?
111. Какой конечный автомат считается недетерминированным? Какую особенность имеет функция переходов недетерминированных конечных автоматов?
112. Как строится диаграмма процессов для недетерминированного конечного автомата? Приведите пример.
113. Как реализуются недетерминированные конечные автоматы?
114. Опишите общую схему преобразования недетерминированного конечного автомата в детерминированный.
115. Какие состояния конечного автомата считаются недостижимыми? Чем эти состояния мешают? Как можно по графу автомата определить наличие недостижимых состояний?
116. Опишите способ удаления недостижимых состояний конечного автомата.
117. В чем заключается минимизация конечного автомата?
118. Какие состояния автомата считаются неразличимыми? Как формируются классы эквивалентных состояний?
119. Опишите общую схему минимизации конечного автомата.
120. В чем принципиальное отличие клеточных автоматов от остальных моделей абстрактных автоматов?
121. Опишите общее устройство клеточного автомата. Что называется элементарным автоматом?
122. Что представляет собой решетка клеточного автомата? Приведите примеры типовых видов решеток.
123. Что называется окрестностью элементарного автомата? Какие виды окрестностей считаются стандартно используемыми?
124. Что представляет собой конфигурация клеточного автомата?
125. Опишите общую схему работы клеточного автомата.
126. Что является объектом изучения при использовании клеточных автоматов?
127. Что задает код Вольфрама? Как он определяется?
128. Приведите формальное определение клеточного автомата.
129. Приведите примеры практического применения клеточных автоматов.

Контрольные вопросы к главе 18

1. Какой алгоритм можно считать правильным?
2. В чем состоит проблема завершаемости алгоритма?
3. Чем отличается понятие «правильный алгоритм» от понятия «корректный алгоритм»?
4. Что понимается под спецификацией алгоритма? Зачем она нужна?
5. Чем отличаются формальные спецификации от обычных?
6. Для чего проводится верификация?
7. Чем отличается формальная верификация?
8. Что называется высказыванием. Приведите примеры высказываний.
9. Что называется предикатом? Приведите примеры различных предикатов.
10. Чем предикаты отличаются от высказываний? Как из предиката получить высказывание?
11. Как классифицируются предикаты?
12. Какие предикаты считаются тождественно истинными, общезначимыми, тождественно ложными, выполнимыми?
13. Что такое квантор? Какие бывают кванторы? Приведите примеры кванторов.
14. Какие переменные в предикате считаются связанными, а какие — свободными?
15. Какие конструкции алгоритмов соответствуют предикатам?
16. Что называется состоянием алгоритма? Что представляет собой пространство состояний алгоритма?
17. Как происходит изменение состояний алгоритма?
18. Как связаны предикаты с пространством состояний алгоритма?
19. Какие состояния алгоритма удовлетворяют тождественно истинному и тождественно ложному предикатам?
20. Как формируются входные и выходные спецификации алгоритмов?
21. Для чего нужны верификационные утверждения?
22. Как включаются спецификации и верификационные утверждения в текст алгоритма?
23. Что представляет собой логика Хоара?
24. Дайте математическое определение тройки Хоара. Для чего она используется?
25. Какой смысл имеет тройка Хоара?
26. Для чего нужна аксиоматическая семантика. Чем она отличается от операционной семантики?
27. Как производится верификация алгоритма? Опишите ее общую схему.
28. Опишите операцию подстановки. Когда она выполнима? Что считается результатом операции подстановки, если она невыполнима?
29. Как задается аксиоматическая семантика оператора присваивания?
30. Приведите примеры, иллюстрирующие аксиому присваивания.
31. Почему верификация алгоритма выполняется в порядке, обратном следованию операторов в алгоритме?

32. Сформулируйте правило функциональной композиции. Приведите примеры применения этого правила.
33. Докажите корректность алгоритма обмена 8.1.
34. Докажите правильность алгоритма 8.2 решения системы двух уравнений с двумя неизвестными. Почему этот алгоритм нельзя считать корректным?
35. Почему в спецификациях алгоритма 8.1 используются вводимые значения, а в спецификациях алгоритма 8.2 они отсутствуют? Какой вариант построения спецификаций следует использовать в общем случае?
36. Сформулируйте правило вывода для управляющей конструкции ветвления.
37. Как на практике используется правило вывода для доказательства корректности условного оператора?
38. Докажите корректность алгоритма 8.3.
39. Докажите корректность алгоритма 8.4.
40. Что называется инвариантом цикла? Какие алгоритмы считаются частично корректными и полностью корректными?
41. В каком порядке доказываются верификационные утверждения для оператора цикла в форме с предусловием?
42. Докажите корректность алгоритма 9.8 вычисления суммы.
43. Как доказывается завершаемость цикла?
44. Опишите общий порядок действий в процессе верификации цикла.
45. Докажите корректность алгоритма 9.1.1 нахождения чисел Фибоначчи.
46. Докажите корректность алгоритма 10.1 определения экстремального элемента массива.
47. Выполните верификацию алгоритма классического поиска.
48. Выполните верификацию алгоритма быстрого поиска.
49. Выполните верификацию алгоритма поиска с барьером.
50. Выполните верификацию алгоритма бинарного поиска.
51. Выполните верификацию алгоритма сортировки выбором.
52. Выполните верификацию алгоритма сортировки вставками.
53. Выполните верификацию алгоритма обменной сортировки.

Контрольные вопросы и упражнения к главе 19

1. Что понимается под вычислительными ресурсами? Какие ресурсы и почему считаются основными?
2. Почему вычислительные ресурсы не всегда могут быть выделены выполняющейся программе?
3. В каких случаях недопустимо недостаточное выделение ресурса процессорного времени?
4. Что можно предпринять, если вычислительных ресурсов не хватает?
5. Какой из алгоритмов, предназначенных для решения одной и той же задачи, считается лучшим?

6. Дайте определения понятий «вход алгоритма», «трудоемкость алгоритма», «функция трудоемкости», «функция объема памяти».
7. Какие характеристики входов алгоритмов влияют на функции трудоемкости и объема памяти? Приведите примеры.
8. Почему вводится понятие «характерный размер входа»? Какие преимущества возникают от его использования?
9. Как выбирается размер входа для практических и теоретических целей?
10. Какие сложности возникают при определении трудоемкости алгоритма?
11. Что называется моделью вычислений? Для чего она вводится? Охарактеризуйте основные модели вычислений.
12. Для чего вводится характеристика алгоритма «вычислительная сложность»?
13. Перечислите и охарактеризуйте основные задачи теории вычислительной сложности.
14. Какими конструкциями алгоритма определяется его вычислительная сложность?
15. Как определить трудоемкость вычисления значения выражения. Приведите примеры.
16. Как определяется трудоемкость обращения к подпрограмме? Приведите примеры.
17. Как рассчитывается трудоемкость оператора присваивания? Приведите примеры.
18. Как определяется трудоемкость управляющей конструкции следования?
19. Почему невозможно определить точную трудоемкость управляющей конструкции ветвления? Какие виды трудоемкости определяются для этой конструкции? Как их находят?
20. Почему невозможно определить точную трудоемкость управляющей конструкции цикла? В каких случаях можно найти трудоемкость цикла стандартными способами? Как вычисляется трудоемкость цикла при использовании стандартного способа?
21. Какие сложности возникают при определении трудоемкости рекурсивных подпрограмм?
22. Опишите стандартный способ определения трудоемкости рекурсивных подпрограмм. Приведите пример его применения.
23. Что требуется для получения однозначного решения рекуррентного уравнения?
24. Опишите метод прямого исключения рекурсии. Продемонстрируйте его применение на примере.
25. Какие рекурсивные соотношения и рекуррентные последовательности связаны с анализируемой рекуррентной подпрограммой?
26. Какие практические результаты можно получить, применяя теорию линейных рекуррентных уравнений?
27. Опишите два способа использования рекуррентного соотношения между элементами последовательности.
28. Что называется линейным неоднородным рекуррентным уравнением? Как определяется его порядок? Какими могут быть коэффициенты этого уравнения в случае анализа трудоемкости алгоритмов?
29. Чем однородное уравнение отличается от неоднородного?
30. Чем уравнения с постоянными коэффициентами отличаются от общего случая?
31. Что называется задачей с начальными условиями для рекуррентного уравнения?

32. Как решаются однородные уравнения первого порядка?
33. Зачем используется характеристическое уравнение при обсуждении рекуррентного уравнения?
34. Как получить характеристическое уравнение для заданного рекуррентного уравнения?
35. Что называется общим решением рекуррентного уравнения?
36. Как получить общее решение рекуррентного уравнения, если характеристическое уравнение имеет разные действительные корни? Приведите примеры.
37. Как получить общее решение рекуррентного уравнения, если характеристическое уравнение имеет одинаковые действительные корни? Приведите примеры.
38. Как получить общее решение рекуррентного уравнения, если характеристическое уравнение имеет комплексные корни? Приведите примеры.
39. Для чего применяется формула Бине?
40. Найдите общее решение уравнения $T(n) = 2T(n-1) + 5$.
41. Как найти решение характеристического уравнения порядка p ? Приведите примеры.
42. Как найти решение рекуррентного уравнения порядка p ?
43. Найдите общее решение для рекуррентного уравнения с характеристическим уравнением $(r-1)^3(r-2)^5(r+3)^4 = 0$.
44. Что называется частным решением неоднородного уравнения? Как находят эти решения?
45. Что считается общим решением неоднородного уравнения? Как находят эти решения?
46. Найдите общее решение уравнения $T(n) = 4T(n-1) - 4T(n-2) + n^2$.
47. Найдите общее решение уравнения $T(n) = -T(n-1) + 2T(n-2) + 2^n$.
48. Найдите общее решение уравнения $T(n) = 6T(n-1) + 9T(n-2) + (-2)^n$.
49. Что считается решением начальной (краевой) задачи для неоднородного рекуррентного уравнения?
50. Решите уравнение $T(n) = 5T(n-1) - 6T(n-2)$ с начальными условиями $T(1) = 2$ и $T(2) = 10$.
51. Решите уравнение $T(n) = 2\sqrt{2}T(n-1) - 4T(n-2)$ с начальными условиями $T(1) = 1$ и $T(2) = 2$.
52. Используя методы решения рекуррентных уравнений, найдите сумму $1^2 + 2^2 + \dots + n^2$.
53. Используя методы решения рекуррентных уравнений, найдите сумму $1^3 + 2^3 + \dots + n^3$.
54. Используя методы решения рекуррентных уравнений, найдите сумму $1 + 4 + 7 + \dots + 2n - 2$.
55. Используя методы решения рекуррентных уравнений, найдите сумму $2 \cdot 1^2 + 3 \cdot 2^2 + \dots + (n+1)n^2$.
56. Какие виды функций трудоемкости обычно получают при анализе алгоритмов? На какие две группы можно разделить эти функции? Чем они принципиально отличаются друг от друга?
57. Что называется классом сложности алгоритмов? Как выделяют основные классы сложности алгоритмов?

58. В чем основная причина появления алгоритмов с экспоненциальной сложностью?
59. Какие действия следует предпринять, если обнаружилось, что построенный алгоритм имеет экспоненциальную сложность?
60. В каких случаях необходимо получить точный вид функции трудоемкости, а в каких можно обойтись приближенной оценкой?
61. Какие возможности предоставляет теория асимптотического анализа алгоритмов? Что находят в процессе такого анализа? При каком условии справедливы его выводы?
62. Что подразумевается под термином «порядок роста»? Как определяется скорость увеличения значения функции в асимптотическом анализе?
63. Какие функции выступают в качестве образцов скорости роста?
64. Приведите применяющиеся в асимптотическом анализе определения мажоранты и миноранты.
65. Приведите формальное определение понятия « O большое». Для чего оно используется в асимптотическом анализе?
66. Назовите возможные смыслы записи $f(n) = O(g(n))$. Что означает эта запись с формальной математической точки зрения?
67. Назовите возможные смыслы записи $O(g(n))$.
68. Дайте определение оборотов «верхняя асимптотическая оценка», «временная асимптотическая сложность», «объемная сложность есть $O(g(n))$ », «мультипликативная константа».
69. Как на практике получаются асимптотические оценки?
70. Чем отличается O большое от o малого?
71. Как обозначается нижняя асимптотическая оценка функций сложности?
72. Чем является запись $f(n) = \Omega(g(n))$ с математической точки зрения?
73. Какая связь существует между O большим и Ω большим?
74. Какие функции считаются асимптотически равными?
75. Чем является запись $f(n) = \Theta(g(n))$ с математической точки зрения?
76. Охарактеризуйте все смысловые особенности обозначения $O(g(n))$ и соотношения $T(n) = O(g(n))$.
77. Какой смысл имеет использование асимптотических обозначений в выражениях и формулах.
78. Какие модели вычислений называют детерминированным вычислителем? Какая модель называется недетерминированным вычислителем?
79. Перечислите свойства асимптотических обозначений, которые используются для получения оценок сложностей алгоритмов.
80. Сформулируйте правило получения верхних асимптотических оценок линейных алгоритмов и алгоритмов с ветвлениями. Приведите примеры.
81. Сформулируйте правило получения верхних асимптотических оценок алгоритмов, содержащих простые циклы. Приведите примеры.
82. Сформулируйте правило получения верхних асимптотических оценок алгоритмов, содержащих кратные циклы. Приведите примеры.

83. Сформулируйте правило получения верхних асимптотических оценок для рекурсивных подпрограмм, полученных разбиением на подзадачи.
84. Приведите типичную шкалу оценок асимптотической сложности алгоритмов.
85. Какие особенности имеет использование асимптотических оценок сложности для малых характерных размеров входов?
86. Какие особенности имеет анализ численных алгоритмов?
87. Зачем потребовалось введение понятия «сложность задачи»? Как она определяется?
88. Какое практическое значение имеет определение класса сложности задачи?
89. Как на практике определяют сложность задачи?
90. Приведите примеры эталонных задач, которые используются для анализа сложности задач?
91. Сформулируйте задачу выполнимости. Какую сложность имеет эта задача?
92. Какие частные случаи задачи выполнимости рассматриваются в теории вычислительной сложности?
93. Какую роль играет задача выполнимости в теории вычислительной сложности?
94. Сформулируйте задачу коммивояжера. Какую сложность она имеет?
95. Сформулируйте переборную задачу в общей постановке. Какую сложность она имеет?
96. В чем состоит задача о рюкзаке?
97. Сформулируйте задачу о ханойских башнях. Какое рекурсивное решение она имеет? Как определяется сложность этой задачи?
98. Что такое сводимость задачи? Из каких составляющих складывается сложность задачи сведения? Какая сводимость называется полиномиальной?
99. Охарактеризуйте классы сложности задач P и EXP. Приведите примеры.
100. Охарактеризуйте класс сложности NP
101. Какие задачи считаются NP-полными?
102. В чем состоит теоретическая важность класса NPC?
103. Почему проблема равенства классов P и NP считается одной из самых сложных?

Контрольные вопросы и упражнения к главе 20

1. Дайте определение понятия «формальный язык». Для чего нужны формальные языки?
2. Что называется цепочкой? Какими характеристиками они описываются? Что такое пустая цепочка? Какие цепочки считаются равными?
3. Дайте определение операции «обращение цепочки» и опишите ее свойства.
4. Дайте определение операции «конкатенация цепочек» и опишите ее свойства.
5. Дайте определение операции «конкатенация множеств» и опишите ее свойства.
6. Дайте определение операции «итерация цепочки» и опишите ее свойства.
7. Дайте определение операции «итерация множества» и опишите ее свойства.

8. Дайте определение операций «звезда Клини» и «плюс Клини». Чем они отличаются друг от друга?
9. Дайте определение операции разбиения и опишите ее свойства. Что называется подцепочкой, префиксом и суффиксом цепочки?
10. Дайте определение операции подстановки. Что такое продукция и чем она отличается от подстановки? Опишите важные частные случаи выполнения операции подстановки.
11. Когда невозможно выполнить подстановку? Что считается результатом операции в этом случае?
12. Какую практическую роль играет подстановка в различных сферах ее использования?
13. Что понимается под выводом цепочки? Дайте определение отношения непосредственной выводимости. Какими свойствами оно обладает?
14. Чем отличается отношение выводимости? Как оно связано с отношением непосредственной выводимости? Что считается шагом вывода?
15. Какие множества считаются регулярными? Дайте рекурсивное определение регулярного множества.
16. Как связаны регулярные множества и регулярные выражения? Какими свойствами обладают регулярные выражения?
17. Дайте определения понятий «схема алгоритма», «простая подстановка», «заключительная подстановка».
18. Опишите порядок действий при выполнении нормального алгоритма. Как описываются эти действия с точки зрения итераций в циклическом алгоритме?
19. Когда нормальный алгоритм не может быть применен к исходной цепочке?
20. Сформулируете данное В. М. Глушковым общее определение алгоритма.
21. Сравните между собой определения алгоритма в теориях частично рекурсивных функций, машин Тьюринга и нормальных алгоритмов Маркова.
22. Что представляет собой язык регулярных выражений? Чем принципиально отличается язык регулярных выражений от алгоритмических языков?
23. Дайте определения терминов «целевой текст», «шаблон», «вхождение», «метасимвол».
24. Перечислите основные возможности поиска в цепочках, которые предоставляет язык регулярных выражений.
25. Для чего требуется экранирование метасимволов? Каким образом оно выполняется?
26. Опишите стандартную классификацию метасимволов.
27. Для чего в шаблонах используются литеральные знаки и строковые литералы? Приведите примеры.
28. Чем отличается символьный набор от строкового литерала? В каких случаях целесообразно использовать символьный набор? Приведите примеры.
29. Опишите существующие способы сокращения записи символьных наборов.
30. Для чего используются символьные сокращения? Приведите примеры типичных символьных сокращений.
31. Когда удобно применять группу? Что представляет собой альтернатива?

32. Для чего в шаблонах применяются квантификаторы? Запишите общий вид и частные случаи квантификатора и приведите примеры их применения.
33. Какие метасимволы-квантификаторы можно использовать в шаблонах и как они соотносятся с общим видом квантификатора?
34. Дайте определение понятия «язык» как средства обмена сообщениями. Как определяются языки в общем случае?
35. Что представляет собой грамматика языка? Определите понятия «лексема», «лексика», «синтаксис».
36. Опишите классификацию языков. Охарактеризуйте различные группы языков.
37. Почему естественные языки не могут использоваться для написания компьютерных программ?
38. Что требуется задать, чтобы полностью описать язык программирования?
39. Для чего применяются метаязыки?
40. Охарактеризуйте метаязык БНФ. Какую структуру имеет БНФ?
41. Что такое нетерминал? Чем отличаются друг от друга терминальный и нетерминальный алфавиты?
42. Опишите и сравните между собой два способа построения имени по одной и той же БНФ.
43. Чем отличается метаязык РБНФ от метаязыка БНФ?
44. Опишите метаязык синтаксических диаграмм Вирта.
45. Сравните между собой определения вещественного числа с помощью языков БНФ, РБНФ и диаграмм Вирта.
46. Чем принципиально отличаются формальные языки от любых других языков? Какую роль они играют в информатике? Приведите примеры формальных языков.
47. Дайте наиболее общее определение языка. Какие языки считаются эквивалентными?
48. Перечислите и охарактеризуйте все возможные способы задания формальных языков.
49. Зачем нужны распознаватели? Чем отличается порождающая грамматика от распознавателя?
50. Сформулируйте строгое определение понятия «продукция». Как связаны продукции формальных грамматик с БНФ?
51. Какие требования предъявляются к цепочкам левой и правой частей продукции? Почему эти требования должны удовлетворяться?
52. Сформулируйте строгое определение формальной грамматики. Рассмотрите пример такой грамматики.
53. Что такое непосредственная выводимость? Определите понятия «шаг вывода», «цепочка вывода». Какая цепочка считается выводимой для данной грамматики?
54. Чем отличается сентенциальная форма от конечной цепочки вывода?
55. Как определяется язык, порождаемый некоторой грамматикой?
56. Какие выводы считаются левосторонним и правосторонним?
57. Что называется деревом вывода? Как оно строится?
58. Какие грамматики считаются эквивалентными? Приведите примеры.
59. Какие грамматики считаются однозначными? Зачем грамматике однозначность?

60. Охарактеризуйте общую структуру классификации грамматик, предложенную Н. Хомским. По какому критерию грамматики относятся к тому или иному типу?
61. Какие грамматики относятся к типу 0? Чем отличаются языки с фразовой структурой? Что считается распознавателем цепочек такого языка?
62. Какие грамматики считаются контекстно-зависимыми? Что такое контекст? Какие автоматы являются распознавателями цепочек контекстно-зависимого языка?
63. Где применяются контекстно-зависимые грамматики и языки? Являются ли языки программирования контекстно-зависимыми?
64. Какие ограничения накладываются на продукции контекстно-свободных грамматик? Какие автоматы являются распознавателями цепочек контекстно-свободного языка?
65. Какие ограничения накладываются на продукции регулярных грамматик? Какие автоматы являются распознавателями цепочек регулярного языка? Приведите примеры продукции регулярной грамматики.
66. Какие грамматики считаются автоматными? Приведите примеры продукции автоматной грамматики? Чем продукция автоматных грамматик отличается от продукции регулярных грамматик?
67. Сформулируйте правила определения типа грамматики по виду ее продукции.
68. Что представляет собой транслятор для языка программирования? Чем отличается компилятор от транслятора?
69. Что является входом и выходом трансляторов и компиляторов?
70. Чем отличается интерпретация от компиляции? Приведите примеры применения интерпретаторов.
71. Охарактеризуйте этапы анализа и синтеза трансляции. Что является выходом этапа анализа? Что формируется на этапе синтеза?
72. Охарактеризуйте фазы, относящиеся к этапу анализа.
73. Охарактеризуйте фазы, относящиеся к этапу синтеза.
74. Опишите основные задачи, решаемые во время выполнения фазы лексического анализа. Что является основой для построения сканера?
75. Опишите основные задачи, решаемые во время выполнения фазы синтаксического анализа. Что является основой для построения парсера?
76. Охарактеризуйте основные задачи трансляции, решаемые на этапе синтеза.
77. Что такое проход трансляции? Какие трансляторы лучше, однопроходные или многопроходные?
78. Чем отличаются ассемблеры от компиляторов?
79. Какую роль играет таблица идентификаторов в процессе трансляции? Опишите ее структуру.
80. Перечислите типичный набор лексем языков программирования.
81. Что является входом и выходом сканера?
82. Чем отличается таблица, содержащая поток лексем, от таблицы идентификаторов?
83. Как заполняются таблицы лексем и идентификаторов?
84. Охарактеризуйте основные задачи сканера.
85. Как сканер определяет границы лексем?

86. Сколько программных реализаций конечных автоматов должно входить в сканер?
87. Сформулируйте правила построения регулярных выражений, эквивалентных отдельным продукциям грамматики.
88. Сформулируйте правила построения совокупности регулярных выражений, эквивалентных регулярной грамматике.
89. Опишите порядок решения системы линейных уравнений, коэффициенты которой являются регулярными выражениями.
90. Опишите порядок действий по получению программной реализации конечного автомата, эквивалентного заданной регулярной грамматике.
91. Опишите порядок построения эквивалентной автоматной грамматики по заданной регулярной грамматике.
92. Какие проблемы могут возникнуть во время преобразования регулярной грамматики в автоматную? Как они решаются?
93. Опишите порядок построения конечного автомата, эквивалентного автоматной грамматике.
94. Какими свойствами должен обладать конечный автомат, чтобы его программная реализация была эффективной и надежной?
95. Сформулируйте основные положения автоматного программирования, которое используется для получения программной реализации конечного автомата.
96. Охарактеризуйте основные задачи синтаксического анализатора и сравните их с основными задачами сканера.
97. Что является входом и выходом парсера?
98. Чем отличается расширенный МП-автомат от обычного МП-автомата?
99. Опишите проблемы, возникающие при программной реализации МП-автоматов, служащих для распознавания цепочек контекстно-свободных грамматик.
100. Какая сложность программных реализаций МП-автоматов считается приемлемой при построении трансляторов?
101. Охарактеризуйте существующие группы распознавателей для контекстно-свободных языков.
102. Какие задачи решаются во время подготовки исходной грамматики к построению программной реализации распознавателя?
103. Какие грамматики называются приведенными? В чем их преимущество перед остальными контекстно-свободными грамматиками?
104. Какие знаки считаются бесплодными? Как они исключаются из грамматики?
105. Какие знаки считаются недостижимыми? Как они исключаются из грамматики?
106. Чем опасны λ -правила и цепные правила? Как они исключаются из грамматики?
107. Как леворекурсивные продукты исключаются из грамматики?
108. Опишите общую схему работы нисходящего распознавателя с подбором альтернатив и возвратами.
109. На чем основана программная реализация нисходящего распознавателя с возвратами?

110. Опишите общий порядок анализа цепочки в программной реализации нисходящего распознавателя с возвратами.
111. Опишите действия, выполняемые на шаге разрастания программной реализации нисходящего распознавателя с возвратами.
112. Опишите действия, выполняемые на шаге успешного сравнения программной реализации нисходящего распознавателя с возвратами.
113. Опишите действия, выполняемые на шаге выбора альтернативы программной реализации нисходящего распознавателя с возвратами.
114. Опишите действия, выполняемые на шаге завершения программной реализации нисходящего распознавателя с возвратами.
115. Опишите общую идею метода рекурсивного спуска. Для каких грамматик он может применяться?
116. Какие грамматики относятся к $LL(k)$ -грамматикам? Опишите общую идею построения распознавателей для языков, описываемых $LL(k)$ -грамматиками.
117. Опишите общую идею построения восходящих распознавателей.
118. Какие грамматики относятся к $LR(k)$ -грамматикам? Опишите общую идею построения распознавателей для языков, описываемых $LR(k)$ -грамматиками.

Контрольные вопросы и упражнения к главе 21

1. Какую роль играет тактовый генератор в работе компьютера?
2. Дайте определения понятий «такт» и «тактовая частота».
3. Какие устройства относятся к дискретным?
4. Что называется вентилем?
5. Охарактеризуйте достоинства и недостатки релейно-контактных вентиляей.
6. Изобразите схемы релейно-контактных вентиляей «И» и «ИЛИ» и опишите их возможные состояния.
7. Нарисуйте схему транзистора и опишите его закрытое и открытое состояния.
8. Нарисуйте схему и условное обозначение базовых вентиляей «НЕ», «НЕ И», «НЕ ИЛИ», «И», «ИЛИ», а также опишите их работу.
9. Опишите алгоритм построения дизъюнктивной нормальной формы.
10. Нарисуйте схему и условное обозначение вентиля «исключающее ИЛИ» и опишите его работу.
11. Каким образом можно реализовать многоходовые вентиляи? Для чего они нужны?
12. Постройте предикаты, описывающие работу полусумматора, и нарисуйте его комбинационную схему.
13. Постройте предикаты, описывающие работу полного сумматора, и нарисуйте его комбинационную схему.
14. Нарисуйте комбинационную схему сдвига кода и опишите ее действие.
15. Постройте комбинационные схемы для предикатов $(\neg a \vee b) \wedge (a \vee c \wedge d)$, $\neg a \wedge \neg b \vee \neg(a \wedge c) \wedge \neg a \wedge b$, $\neg(\neg a \wedge b \vee \neg b \wedge c) \vee a \wedge \neg d$.

16. Для чего используется компаратор? Нарисуйте его комбинационную схему и опишите принцип действия.
17. Для чего используется декодер? Нарисуйте его комбинационную схему и опишите принцип действия.
18. Для чего используется мультиплексор? Нарисуйте его комбинационную схему и опишите принцип действия.
19. Дайте определения понятий «последовательный код» и «параллельный код». Опишите основанный на использовании мультиплексора способ преобразования параллельного кода в последовательный.
20. Нарисуйте схему одноразрядного арифметико-логического устройства и опишите принцип его действия.
21. Чем отличаются комбинационные схемы от схем с памятью (конечных автоматов)?
22. Что представляет собой триггер? Нарисуйте его схему.
23. Сколько возможных состояний имеет триггер? Какие состояния триггера считаются устойчивыми, а какие неустойчивыми?
24. Опишите поведение триггера при записи в него единицы для всех возможных исходных состояний.
25. Опишите поведение триггера при записи в него нуля для всех возможных исходных состояний.
26. Опишите особенности D -триггера. Зачем понадобилось изменять основную схему триггера?
27. Что представляет собой интегральная схема? Нарисуйте простой вариант универсальной интегральной схемы.
28. Опишите программную модель оперативной памяти.
29. Для чего требуется выравнивать данные по границам слов памяти?
30. Для чего потребовалось вводить сегментацию оперативной памяти? Что такое сегмент памяти? Чем он характеризуется? Опишите организацию сегментной адресации.
31. Что представляет собой регистровый уровень памяти?
32. Сформулируйте основные различия между регистровым уровнем памяти и оперативной памятью.
33. Перечислите основные группы регистров процессора, укажите назначение и особенности их структуры.
34. Опишите механизм формирования физического адреса в процессоре.
35. Опишите общую структуру машинной команды.
36. Что определяет адресность машинной команды? Опишите особенности машинных команд различной адресности.
37. Охарактеризуйте применяющиеся в машинных командах способы адресации.
38. Сравните между собой машинный и ассемблерный форматы команды.
39. Опишите алгоритм исполнения машинной команды процессором компьютера.
40. Обоснуйте тезис о необходимости широкого использования параллелизма в компьютерных системах.
41. Чем отличается собственно параллелизм от конвейеризации?

42. Охарактеризуйте отличительные особенности начального периода электронного этапа в развитии вычислительных систем.
43. Изложите базовые принципы организации вычислительных систем, выдвинутые Дж. фон Нейманом. Изобразите структурную схему машины фон Неймана.
44. Опишите принципы организации памяти на ртутных линиях задержки и ферритовых кольцах. Чем отличается разрядно-последовательная обработка кодов от разрядно-параллельной?
45. Какую роль играют каналы в работе компьютеров?
46. Для чего нужны контроллеры? В чем отличие контроллеров от каналов? Чем отличаются контроллеры от адаптеров?
47. Опишите принцип работы контроллеров DMA.
48. Для чего нужен буфер? Какими бывают буферы?
49. Сформулируйте изложенные академиком С. А. Лебедевым идеи по развитию архитектуры вычислительных систем.
50. Охарактеризуйте микроархитектурный уровень структуры процессора.
51. Поясните смысл терминов «микрокоманда» и «микропрограмма».
52. Какие микрокоманды можно выделить в реализации команды умножения?
53. Опишите конвейерный режим функционирования процессора.
54. Поясните смысл терминов «конвейер», «ступень конвейера», «длина конвейера».
55. Оцените возможное ускорение работы процессора от использования конвейера.
56. Какая архитектура считается суперконвейерной?
57. Почему нельзя включить в состав процессора большое количество конвейеров?
58. Опишите суперскалярную архитектуру процессора.
59. Дайте сравнительную характеристику суперконвейерной и суперскалярной архитектур.
60. Какие проблемы возникают при конвейерной организации процессоров?
61. Из каких механизмов складывается динамическое исполнение машинных команд?
62. Какие взаимозависимости существуют между машинными командами программ?
63. Почему машинные команды должны покидать конвейер в том же порядке, в котором поступили на него?
64. В чем состоит метод изменения порядка следования команд?
65. Что дает предсказание ветвления?
66. Сравните обычный и многопоточный способы выполнения машинных команд.
67. Дайте сравнительную характеристику многопроцессорных и многоядерных систем.
68. Охарактеризуйте основные уровни памяти компьютера
69. Охарактеризуйте вспомогательные уровни памяти компьютера.
70. Опишите принцип действия микросхем статической памяти.
71. Опишите принцип действия микросхем динамической памяти.
72. Что представляет собой регенерация памяти? Как и когда она производится?
73. Опишите принципы использования кэша в архитектуре компьютера.

74. Сформулируйте принципы заполнения кэша данными.
75. Поясните смысл терминов «попадание в кэш», «промах кэша», «кэширование».
76. Оцените возможный выигрыш от включения кэша в структуру компьютера.
77. Опишите структуру многоуровневого кэша и дайте характеристику его уровней.
78. Какие микросхемы используются для реализации кэша?
79. Как определяется пропускная способность шины?
80. Опишите принцип работы мультиплексных шин.
81. В чем состоит перекоп шин?
82. Опишите конвейерный режим работы шин.
83. В чем преимущество многосшинной архитектуры компьютера?
84. Охарактеризуйте архитектуру с двойной независимой шиной.
85. Для чего нужен чипсет? Что он определяет? Охарактеризуйте его функции.
86. Поясните смысл терминов «северный мост», «южный мост», «локальная шина», «шина памяти».

Контрольные вопросы к главе 22

1. Почему разработка программного обеспечения считается инженерной дисциплиной?
2. Охарактеризуйте основные задачи, которые решаются программной инженерией.
3. Какие действия предусматривает системный подход к разработке программного обеспечения?
4. Какие вопросы разработки программного обеспечения регламентируются стандартами ЕСПД?
5. Что вкладывается в понятие «качество программного обеспечения»?
6. Охарактеризуйте основные критерии качества программного обеспечения.
7. Что должен содержать пакет документов программной системы?
8. Почему невозможно создание программной системы, не содержащей ошибок?
9. Каким образом решается проблема создания программного обеспечения с учетом неизбежности ошибок?
10. Дайте определение терминов «отказ», «зависание», «сбой».
11. Охарактеризуйте свойства, которыми должно обладать надежное программное обеспечение.
12. Как связаны между собой надежность и стоимость программного обеспечения?
13. Как связаны между собой корректность и надежность программного обеспечения?
14. Что называется жизненным циклом программного обеспечения? Перечислите и охарактеризуйте основные этапы жизненного цикла.
15. Из каких элементов складывается жизненный цикл программного обеспечения?
16. Что называется процессом жизненного цикла программного обеспечения? Охарактеризуйте типичные процессы жизненного цикла.
17. Что включает в себя определение конкретного процесса жизненного цикла программного обеспечения?

18. Дайте определения понятий «вид деятельности», «роль», «артефакт». Приведите примеры видов деятельности, ролей и артефактов, связанных с процессами жизненного цикла программного обеспечения.
19. Охарактеризуйте стандартные основные процессы жизненного цикла программного обеспечения.
20. Охарактеризуйте стандартные, вспомогательные и организационные процессы жизненного цикла программного обеспечения.
21. Какой смысл вкладывается в понятие «спецификация программного обеспечения»?
22. Что называется стадией жизненного цикла программного обеспечения?
23. Охарактеризуйте типичные стадии жизненного цикла программного обеспечения.
24. Как связаны между собой процессы и стадии жизненного цикла программного обеспечения?
25. Что называется моделью жизненного цикла программного обеспечения? Какую роль играют модели жизненного цикла в разработке программного обеспечения?
26. Опишите основные особенности каскадной модели жизненного цикла программного обеспечения. В каких ситуациях целесообразно ее использование?
27. Опишите спиральную модель жизненного цикла программного обеспечения. В каких ситуациях целесообразно ее использование?
28. Сравните между собой каскадную и спиральную модели жизненного цикла программного обеспечения.
29. Что отличает технологии быстрой разработки приложений и экстремального программирования?
30. Что приходится выбирать разработчику программного обеспечения в рамках конкретного жизненного цикла программного обеспечения?
31. Что называется методологией разработки программного обеспечения? Чем определяется конкретный вариант методологии?
32. Что понимается под семейством методологий разработки программного обеспечения?
33. Сравните между собой в принципиальном плане императивную и декларативную методологии разработки программного обеспечения.
34. Сформулируете базовый принцип и опишите базовый метод императивной методологии.
35. Охарактеризуйте особенности процедурного подхода в императивной методологии.
36. Охарактеризуйте структурный подход в императивной методологии. Какую роль играет этот подход в настоящее время?
37. Что понимается под визуальным программированием?
38. Какие особенности имеют универсальные графические языки программирования. Приведите примеры таких языков.
39. Что отличает автоматное программирование от обычного императивного подхода?
40. Для каких классов задач следует выбирать императивную методологию и различные ее подходы?
41. Как воспринимается проблемная область с точки зрения ООП? Приведите пример такого представления предметной области.

42. В чем проявляется принципиальное различие в понимании программы с точки зрения стандартной императивной методологии и с точки зрения методологии ООП?
43. В чем проявляются синтаксические различия в обычных программах и программах, написанных в соответствии с методологией ООП?
44. Сформулируйте базовый принцип методологии ООП.
45. Дайте определения понятий «объект», «класс», «экземпляр объекта». С какими понятиями в традиционных программах можно их сопоставить?
46. Сформулируйте принцип инкапсуляции ООП. Какую роль он играет в разработке программ на основе ООП?
47. Сформулируйте принцип наследования ООП. Какую роль он играет в разработке программ на основе ООП?
48. Дайте определения понятий «родительский класс», «дочерний класс», «метод класса».
49. Сформулируйте принцип полиморфизма ООП. Почему потребовалось добавить этот принцип в методологию ООП?
50. Какие понятия императивной методологии отсутствуют в декларативной методологии?
51. Сформулируйте основной принцип функционального подхода декларативной методологии.
52. Какие конструкции и действия допустимы в функциональных программах?
53. Чем различаются понятия функции в императивной методологии и в функциональном подходе декларативной методологии?
54. Чем вреден побочный эффект в подпрограммах императивного подхода?
55. Охарактеризуйте преимущества и недостатки функционального подхода декларативной методологии.
56. Как описывается проблемная область с точки зрения логического подхода декларативной методологии.
57. Сформулируйте основной принцип логического подхода декларативной методологии.
58. Как понимаются «факт», «отношение» и «правило вывода» с точки зрения логического подхода декларативной методологии?
59. Что может быть элементом списка в логических программах? Что считается головой и хвостом списка?
60. Какую структуру имеет список в логических программах?
61. Что представляет собой программа с точки зрения логического подхода?
62. Охарактеризуйте преимущества и недостатки логического подхода декларативной методологии.
63. Для каких классов задач следует выбирать функциональный и логический подходы декларативной методологии?
64. Что определяют технологии разработки программного обеспечения? Дайте формальное определение этого понятия.
65. Что должна содержать любая технология разработки программного обеспечения?

66. Охарактеризуйте типичный набор стадий, связанных с императивной методологией разработки программного обеспечения.
67. Какие действия необходимо осуществить на стадии формирования требований?
68. Из каких этапов состоит стадия формирования требований?
69. Где используются результаты работы, выполненной на стадии формирования требований?
70. Какую роль играет стадия формирования требований в процессе разработки программного обеспечения?
71. Как классифицируются требования к программной системе?
72. Охарактеризуйте группу функциональных требований.
73. Какие требования могут входить в группу нефункциональных требований?
74. Чем отличаются требования разработчиков от системных требований?
75. Какую роль играет анализ требований в процессе разработки программного обеспечения?
76. Какими свойствами должны обладать требования и их совокупность после завершения анализа требований?
77. Охарактеризуйте существующие способы документирования требований к программной системе.
78. Что называется спецификацией требований к программной системе? Как записывается спецификация требований?
79. Чем отличается техническое задание от спецификации требований?
80. Для чего требуется создание проекта разрабатываемой программной системы?
81. Что определяет стандарт проектирования организации — разработчика программного обеспечения?
82. Какой способ применяется для борьбы со сложностью проектирования программных систем?
83. Какие требования предъявляются к составным частям, которые выделяются в процессе декомпозиции?
84. Сравните между собой роли, которые играют анализ и синтез на стадии проектирования программного обеспечения.
85. Чем отличается декомпозиция структурного подхода от декомпозиции объектного подхода?
86. Охарактеризуйте стандартные этапы стадии проектирования программного обеспечения.
87. Что должна содержать рабочая проектная документация?
88. Какие задачи решает архитектурное проектирование?
89. Что дает использование каркасного подхода в проектировании?
90. Что дает использование паттернов проектирования?
91. Сравните между собой каркасный подход и паттерны проектирования.
92. Какие задачи решает детальное проектирование?
93. Какие средства используются для выполнения детального проектирования?

94. Какие задачи решает проектирование интерфейсов?
95. Сформулируйте основные правила проектирования интерфейсов.
96. Охарактеризуйте основные способы проектирования интерфейсов.
97. В каком порядке осуществляются архитектурное и детальное проектирование и проектирование интерфейсов?
98. Какую роль играют языки моделирования в проектировании программных систем?
99. Для чего используются ER-диаграммы? Как они строятся?
100. Сравните между собой основные стандарты построения ER-диаграмм.
101. Что представляет собой язык моделирования UML? Для чего он используется в проектировании программного обеспечения?
102. Опишите основные элементы диаграммы языка UML.
103. Охарактеризуйте основные виды диаграмм языка UML.
104. Какие средства используются проектировщиками для автоматизации процесса проектирования?
105. Какими свойствами должно обладать современное CASE-средство?
106. Какие задачи решаются на стадии программирования?
107. Как осуществляется выбор языка программирования?
108. Опишите классификацию языков программирования по поколениям.
109. Чем отличается стандарт языка от его реализации?
110. Как классифицируются языки программирования по используемой методологии. Приведите примеры.
111. Опишите классификацию языков программирования по проблемной области.
112. Охарактеризуйте достоинства и недостатки семейства алгоподобных языков.
113. Охарактеризуйте достоинства и недостатки семейства Си-подобных языков.
114. Классифицируйте языки по типизации.
115. Перечислите свойства, которыми должен обладать современный язык программирования.
116. Что понимается под системой программирования? Какую роль играют такие системы в разработке программных систем?
117. Что входит в состав современной системы программирования?
118. Что понимается под стилем программирования?
119. Какие группы правил включает в себя понятие «стиль программирования»?
120. Как рекомендуется выбирать имена в программах и форматировать программные конструкции?
121. Сформулируйте основные рекомендации по стилю программирования?
122. Почему неизбежны ошибки в создаваемой программной системе?
123. Какую роль играет стадия тестирования в разработке программного обеспечения?
124. Дайте определения понятий «верификация», «отладка», «тестирование», «аттестация», «валидация».
125. Чем отличается верификация от аттестации?

126. Как связано тестирование с верификацией и отладкой?
127. Как осуществляется тестирование на компьютере?
128. Что должен включать в себя тестовый набор? Что такое тестовый прогон?
129. Какой тестовый набор считается успешным, а какой — неуспешным?
130. Что понимается под исчерпывающим тестированием и почему такое тестирование невозможно?
131. Сформулируйте общепринятые рекомендации по проведению тестирования.
132. Сравните между собой основные методы тестирования — статический анализ и динамическое тестирование.
133. Каким образом производится статический анализ? Какие задачи он решает?
134. Чем отличаются инспекции от сквозных просмотров? Какого рода ошибки обнаруживают эти методы?
135. Какие типовые вопросы задают при проведении инспекций?
136. Какие ошибки невозможно обнаружить с помощью методов статического анализа?
137. Чем различаются методы белого и черного ящиков?
138. Сравните между собой задачи, решаемые модульным интеграционным и системным тестированием.
139. Какие методы используются при модульном тестировании?
140. Какие методы используются при интеграционном тестировании?
141. Какие методы используются при системном тестировании?
142. Какие варианты покрытий используются в методе белого ящика? Сравните их между собой по эффективности обнаружения ошибок и сложности осуществления.
143. Какие варианты покрытий используются в методе черного ящика?
144. Как формируются классы эквивалентности наборов исходных данных?
145. Опишите два основных класса эквивалентности наборов исходных данных.
146. Что называется робастностью программной системы?
147. В чем сущность анализа граничных значений?
148. Сформулируйте основные рекомендации метода анализа граничных значений по построению тестовых наборов.
149. В каких формах проводится тестирование производительности?

Приложение. Вектора и матрицы

Рассматривая различные величины, легко заметить, что некоторые из них задаются только числовым значением, а для задания других нужно указывать не только значение, но и направление. Величины первой группы относят к скалярам, а второй — к векторам. Например, время и температура — это скалярные величины, а сила и скорость — векторные.

Скалярные величины отождествляются с точками на числовой прямой, и любое действительное число $\alpha \in \mathbb{R}$ считается скаляром. Естественным способом задания вектора (то есть его направления и значения) является указание двух точек, в которых он начинается и заканчивается, что и отражено в его стандартном определении: «Вектором \overline{AB} называется направленный отрезок, идущий из точки A в точку B ».

Такое определение хорошо использовать для чисто геометрических задач. Однако во многих случаях более удобен координатный способ задания векторов. Его можно применять, если введена, например, прямоугольная декартова система координат. Декартовы координаты (a_x, a_y) любой точки A на плоскости можно рассматривать как координаты вектора \overline{a} , идущего из начала системы координат точки O в эту точку, $\overline{a} = OA$. Таким образом, пара чисел (a_x, a_y) полностью и однозначно определяет указанный вектор. Аналогично, тройка чисел (a_x, a_y, a_z) — декартовы координаты некоторой точки в пространстве — может считаться определением вектора \overline{a} , идущего из начала системы координат в эту точку.

Для множества двух- и трехмерных векторов стандартным образом вводятся обладающие обычными свойствами (коммутативность, ассоциативность и т. д.) операции сложения векторов и умножения вектора на число, и говорят, что множество векторов образует **линейное векторное пространство**. Пространство двумерных векторов принято обозначать \mathbb{R}^2 , а пространство трехмерных векторов — \mathbb{R}^3 .

Обобщая эти понятия, набор, состоящий из n действительных чисел (a_1, a_2, \dots, a_n) , можно рассматривать как n -мерный вектор. Однако использовать названия координатных осей x , y и z для обозначения отдельных элементов внутри набора, как это делается в случае плоскости или трехмерного пространства, теперь невозможно. Поэтому числа из набора просто нумеруются a_1, a_2, \dots, a_n и называются **элементами** или **компонентами** вектора.

Если для множества n -мерных векторов по аналогии с \mathbb{R}^2 и \mathbb{R}^3 определить операции сложения векторов и умножения на число, то естественно считать, что такое множество образует n -мерное линейное векторное пространство, и обозначить его \mathbb{R}^n . Очевидно, что пространства \mathbb{R}^2 и \mathbb{R}^3 являются важными частными случаями пространства \mathbb{R}^n .

В любом n -мерном векторном пространстве имеется единственный **нулевой** вектор с компонентами $(0, 0, \dots, 0)$, а также ровно n различных **единичных** векторов $(1, 0, 0, \dots, 0)$; $(0, 1, 0, \dots, 0)$; $(0, 0, 1, \dots, 0)$; $(0, 0, 0, \dots, 1)$.

Основные операции над векторами, принадлежащими \mathbb{R}^n , определяются точно так же, как для плоскости и трехмерного пространства. Так, суммой двух векторов является вектор, каждый элемент которого равен сумме элементов складываемых векторов. Формально это определение можно записать следующим образом. Пусть $\overline{a} = (a_1, a_2, \dots, a_n) \in \mathbb{R}^n$ и $\overline{b} = (b_1, b_2, \dots, b_n) \in \mathbb{R}^n$, тогда суммой векторов \overline{a} и \overline{b} называется вектор $\overline{c} = \overline{a} + \overline{b} \in \mathbb{R}^n$, компоненты (c_1, c_2, \dots, c_n) которого равны $c_i = a_i + b_i$, $i = 1, 2, \dots, n$. Если, например, $\overline{a} = (2, 5, 7, 4)$ и $\overline{b} = (4, 1, 0, -1)$, тогда $\overline{c} = \overline{a} + \overline{b} = (2 + 4, 5 + 1, 7 + 0, 4 - 1) = (6, 6, 7, 3)$.

Произведением вектора \bar{a} на число α является вектор $\bar{c} = \alpha\bar{a} \in \mathbb{R}^n$, каждый элемент (c_1, c_2, \dots, c_n) которого равен произведению этого числа на соответствующие элементы исходного вектора $c_i = \alpha a_i$, $i = 1, 2, \dots, n$. Пусть $\alpha = 3$ и $\bar{a} = (2, 5, 7, 4)$, тогда $\bar{c} = \alpha\bar{a} = (3 \cdot 2, 3 \cdot 5, 3 \cdot 7, 3 \cdot 4) = (6, 15, 21, 12)$.

Если взять любые два вектора, \bar{a} и $\bar{b} \in \mathbb{R}^n$, и любые два числа, α и $\beta \in \mathbb{R}$, то выражение $\alpha\bar{a} + \beta\bar{b}$ также является вектором, который принадлежит \mathbb{R}^n . Такое выражение называется **линейной комбинацией** исходных векторов. Собственно говоря, векторное пространство называется линейным именно потому, что любая линейная комбинация векторов, принадлежащих рассматриваемому пространству, принадлежит этому же пространству.

В векторном пространстве можно выделить подмножество векторов, любая линейная комбинация которых принадлежит этому же подмножеству. Такое подмножество векторов называется **подпространством** векторного пространства. Например, в трехмерном пространстве множество векторов, лежащих в плоскости OXY , образует двухмерное подпространство.

Если компоненты векторов и число α принадлежат множеству действительных чисел, а сложение и умножение определены обычным способом, то говорят, что линейное векторное пространство \mathbb{R}^n **задано над полем** действительных чисел \mathbb{R} .

ВНИМАНИЕ! Выражение «задано над полем» указывает, какими числами разрешено пользоваться, множество, которому могут принадлежать рассматриваемые элементы.

Важным для помехоустойчивого кодирования случаем является задание векторного пространства над полем Галуа $\text{GF}(2)$, когда компоненты векторов могут принимать только два значения, 0 или 1, а операции сложения и умножения выполняются по модулю 2. Пусть, например, $\bar{x} = (1, 0, 1, 1)$, $\bar{y} = (0, 1, 1, 1)$. Их суммой является вектор $\bar{z} = \bar{x} + \bar{y} = (1 \oplus 0, 0 \oplus 1, 1 \oplus 1, 1 \oplus 1) = (1, 1, 0, 0)$ — результат также состоит из компонент со значением 0 либо 1.

В основном тексте книги в разделе 9.2.1.6 обсуждается задача 9.3 о скалярном умножении двух векторов. Геометрически эта операция определяется как произведение длин векторов на косинус угла между ними. Скалярное умножение широко используется при решении различных задач. Например, из равенства нулю скалярного произведения векторов следует, что эти вектора взаимно перпендикулярны.

Существует и более удобное определение скалярного произведения векторов через их компоненты: скалярным произведением $(\bar{a}\bar{b})$ векторов $\bar{a} = (a_1, a_2, \dots, a_n)$ и $\bar{b} = (b_1, b_2, \dots, b_n)$ называется число p , равное сумме произведений соответствующих компонент векторов \bar{a} и \bar{b} :

$$p = (\bar{a}\bar{b}) = \sum_{k=1}^n a_k b_k = a_1 b_1 + a_2 b_2 + \dots + a_n b_n.$$

Пусть вновь $\bar{a} = (2, 5, 7, 4)$ и $\bar{b} = (4, 1, 0, -1)$, тогда $(\bar{a}\bar{b}) = 2 \cdot 4 + 5 \cdot 1 + 7 \cdot 0 - 4 \cdot 1 = 7$. Рассмотрим также пример нахождения скалярного произведения для векторов, заданных над полем Галуа $\text{GF}(2)$. Если взять те же вектора $\bar{x} = (1, 0, 1, 1)$ и $\bar{y} = (0, 1, 1, 1)$, то их скалярное произведение окажется равным $(\bar{x}\bar{y}) = 1 \otimes 0 \oplus 0 \otimes 1 \oplus 1 \otimes 1 \oplus 1 \otimes 1 = 0 \oplus 0 \oplus 1 \oplus 1 = 0$.

В ряде задач приходится рассматривать совокупности из нескольких векторов $\vec{a}_1, \vec{a}_2, \dots, \vec{a}_m$. Тогда для одновременного представления компонент всех этих векторов удобно использовать таблицу вида

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \dots & \dots & \dots & \dots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix},$$

в которой элементы каждого вектора занимают отдельную строку. В такой совокупности каждый элемент приходится обозначается двумя номерами, которые принято называть **индексами**. Первый индекс всегда обозначает вектор (строку таблицы), а второй — его элемент (столбец таблицы).

ВНИМАНИЕ! Пара индексов является упорядоченной — сначала всегда указывается номер строки, а потом — номер столбца.

Задав оба индекса, можно однозначно указать нужный элемент: он находится на пересечении строки и столбца с заданными номерами (индексами). Например, a_{12} — это второй элемент первого вектора \vec{a}_1 , а a_{21} — это первый элемент второго вектора. Видно, что a_{12} и a_{21} — это разные по смыслу величины, хотя по значению они могут и совпадать. Такого рода таблицы называются **матрицами**. Матрицы широко применяются для компактной записи различных математических объектов, например систем линейных алгебраических уравнений. В этом случае количество строк матрицы соответствует количеству уравнений, а количество столбцов — количеству неизвестных.

ВНИМАНИЕ! **Матрицей** называется прямоугольная таблица, образованная совокупностью пронумерованных строк и столбцов, на пересечении которых находятся имеющие два номера (два **индекса**) элементы матрицы. **Размерностью** матрицы называется количество строк и столбцов, из которых она состоит.

Размерность матрицы может задаваться прямым текстовым указанием или в виде надстрочного или подстрочного произведения двух целых чисел, которое размещено справа от названия матрицы: $A^{m \times n}$ или $A_{m \times n}$. В таком произведении первое из чисел m определяет количество строк, а второе n — количество столбцов матрицы.

Вместо приведенной выше довольно громоздкой полной формы записи матрицы часто используют эквивалентную компактную форму $A = (a_{ij})$, $i = 1, \dots, m$, $j = 1, \dots, n$.

В этой форме размерность может задаваться справа от обозначения элементов матрицы — $(a_{ij})_m^n$.

В общем случае элементы матрицы могут быть целыми, рациональными, действительными, комплексными числами, функциями, а также некоторыми другими объектами и структурами. Поэтому, определяя матрицу, обычно указывают, какими могут быть ее элементы, какими числами разрешено пользоваться, точнее, над каким полем она задана.

Матрица, у которой количество строк равно количеству столбцов, $m = n$, называется **квадратной**. Такие матрицы характеризуются **порядком**, равным количеству строк и столбцов. Если $m \neq n$, матрица называется **прямоугольной**. При этом возможны оба варианта, и $m < n$, и $m > n$, но в любом случае должно выполняться условие $m \geq 1$ и $n \geq 1$.

Матрица размерности $1 \times n$, $n > 1$, называется **вектор-строкой**, а матрица размерности $m \times 1$, $m > 1$, — **вектор-столбцом**. Вектор-строка и вектор-столбец по своим свойствам соответствуют не только матрицам, но и векторам, поэтому их можно трактовать и использовать в зависимости от необходимости и как матрицу, и как вектор.

Матрица O , все элементы которой равны нулю, называется **нулевой**, а квадратная матрица I , у которой на диагонали находятся 1, а все остальные элементы равны 0, называется **единичной**. Примеры матриц:

$$A^{4 \times 4} = \begin{pmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{pmatrix}; B^{3 \times 4} = \begin{pmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \end{pmatrix}; C^{4 \times 3} = \begin{pmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \\ c_{41} & c_{42} & c_{43} \end{pmatrix};$$

$$D^{1 \times 4} = (d_1 d_2 d_3 d_4); F^{4 \times 1} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{pmatrix}; O^{3 \times 4} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}; I^{4 \times 4} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Матрица $A^{4 \times 4}$ — квадратная, матрицы $B^{3 \times 4}$ и $C^{4 \times 3}$ — прямоугольные, $D^{1 \times 4}$ и $F^{4 \times 1}$ — вектор-строка и вектор-столбец соответственно, матрица $O^{3 \times 4}$ — нулевая, $I^{4 \times 4}$ — единичная.

ВНИМАНИЕ! При записи вектора его элементы принято отделять друг от друга запятой, а при записи матриц между их элементами ничего не ставится.

Например, запись $(2, 5, 7, 4)$ рассматривается как вектор, но $(2 \ 5 \ 7 \ 4)$ — это вектор-строка, то есть разновидность матрицы. В принципе, вектор и вектор-строку можно не различать, так как по сути это один и тот же объект, который в зависимости от обстоятельств, от решаемой задачи трактуется или как вектор, или как матрица. А выбранный способ записи только подчеркивает, как именно трактуется объект.

В матрице можно выделить группу строк и(или) столбцов и рассматривать элементы, попавшие в нее, как самостоятельную матрицу. Такую группу элементов матрицы называют **подматрицей**, **субматрицей** или **блоком** матрицы. Более формальное определение: «Подматрицей размерности $k \times l$ матрицы A размерности $m \times n$ называется совокупность элементов этой матрицы, находящихся на пересечении k подряд идущих строк и l подряд идущих столбцов, $1 \leq k \leq m$, $1 \leq l \leq n$ ». Выделение в матрице любой подматрицы автоматически приводит к появлению в ее остальных частях еще одной или нескольких подматриц. Подматрицы сохраняют все свойства обычных матриц.

Рассмотрим в качестве примера матрицу

$$G^{4 \times 7} = \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 \end{pmatrix}.$$

В ней можно выделить, например, первые четыре столбца и рассматривать их как единичную подматрицу $I^{4 \times 4}$. При этом оставшиеся три столбца образуют подматрицу $P^{4 \times 3}$:

$$G^{4 \times 7} = \left(\underbrace{\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}}_{I^{4 \times 4}} \mid \underbrace{\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}}_{P^{4 \times 3}} \right); \quad I^{4 \times 4} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}; \quad P^{4 \times 3} = \begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}.$$

Разбиение матрицы на подматрицы можно отобразить в компактной форме с помощью записи вида $G^{4 \times 7} = (I^{4 \times 4}, P^{4 \times 3})$ или $G = (I, P)$.

Если в матрице A каждую строку сделать столбцом, то получится другая матрица, в которой столбцов окажется столько же, сколько было строк, а строк столько же, сколько было столбцов в исходной матрице. Такая матрица называется **транспонированной** и обозначается A^T . С формальной точки зрения элементы транспонированной матрицы $A^T = (a_{ij}^T)$ определяются через элементы исходной матрицы $A = (a_{ij})$ соотношениями $a_{ij}^T = a_{ji}$, $i = 1, 2, \dots, m$; $j = 1, 2, \dots, n$. Например,

$$A = \begin{pmatrix} 1 & 3 & 0 & 4 \\ 5 & 3 & 7 & 1 \\ 8 & 0 & 6 & 2 \end{pmatrix}; \quad A^T = \begin{pmatrix} 1 & 5 & 8 \\ 3 & 3 & 0 \\ 0 & 7 & 6 \\ 4 & 1 & 2 \end{pmatrix}.$$

Заметим, что при транспонировании вектор-строки получается вектор-столбец и, наоборот, при транспонировании вектор-столбца получается вектор-строка.

Матрицы одинаковых размерностей можно складывать, при этом получится новая матрица той же самой размерности, ее элементы равны сумме соответствующих элементов матриц слагаемых. Пусть имеются $A^{m \times n}$ и $B^{m \times n}$, тогда $C^{m \times n} = A^{m \times n} + B^{m \times n}$, $c_{ij} = a_{ij} + b_{ij}$, $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$.

Если, например, заданы матрицы $A^{3 \times 4}$ и $B^{3 \times 4}$,

$$A = \begin{pmatrix} 1 & 3 & 0 & 4 \\ 5 & 3 & 7 & 1 \\ 8 & 0 & 6 & 2 \end{pmatrix}; \quad B = \begin{pmatrix} -1 & 2 & 6 & 0 \\ -2 & 5 & 1 & 4 \\ -4 & 3 & 3 & 7 \end{pmatrix},$$

то, выполнив в соответствии с этим определением их суммирование, получим в результате матрицу

$$C = \begin{pmatrix} 1+(-1) & 3+2 & 0+6 & 4+0 \\ 5+(-2) & 3+5 & 7+1 & 1+4 \\ 8+(-4) & 0+3 & 6+3 & 2+7 \end{pmatrix} = \begin{pmatrix} 0 & 5 & 6 & 4 \\ 3 & 8 & 8 & 5 \\ -4 & 3 & 9 & 9 \end{pmatrix}.$$

Матрицу можно умножить на число, при этом получается новая матрица той же размерности, ее элементы равны произведению соответствующих элементов исходной матрицы на это число. Пусть, например, имеются $A^{m \times n}$ и число $\alpha \in \mathbb{R}$, тогда $C^{m \times n} = \alpha A^{m \times n}$, $c_{ij} = \alpha a_{ij}$, $i = 1, 2, \dots, m$, $j = 1, 2, \dots, n$. Легко заметить, что суммирование матриц и умножение матрицы на число выполняются точно так же, как выполняются эти же операции над векторами.

Операция умножения матриц в какой-то мере может рассматриваться как расширение операции скалярного умножения векторов, поскольку матричное умножение выполняется следующим образом:

1. Строки и столбцы матриц сомножителей рассматриваются как вектора.
2. Каждая строка первой матрицы скалярно умножается на каждый столбец второй матрицы.
3. В результате получается очередной элемент матрицы-результата с индексами, равными номерам строки и столбца, которые участвовали в умножении.

Более формально операцию матричного умножения можно определить следующим образом: элементы c_{ij} матрицы результата вычисляются с помощью соотношения

$$c_{ij} = \sum_{k=1}^n a_{ik} b_{kj}, \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n.$$

Напомним, скалярно умножить два вектора можно только в том случае, если они имеют одинаковое количество элементов. Аналогично, на возможность выполнения матричного умножения накладывается жесткое условие: умножать матрицы можно только тогда, когда количество столбцов первой матрицы равно количеству строк второй. Пусть, например, умножаются матрицы $A^{m \times n}$ и $B^{n \times l}$. Видно, что для них необходимое условие умножения выполнено. В результате умножения получится новая матрица $C^{m \times l} = A^{m \times n} \times B^{n \times l}$, у которой столько же строк m , сколько у первой матрицы, а столбцов l столько же, сколько у второй.

Из определения операции умножения матриц следует, что она некоммутативна, то есть $A \times B$ может не совпадать с $B \times A$, и матрицы сомножители менять местами нельзя. Более того, из-за несовпадения количества столбцов первой матрицы с количеством строк второй может оказаться, что нельзя умножать либо A на B , либо B на A .

ВНИМАНИЕ! Порядок следования сомножителей может повлиять не только на результат, но и на возможность выполнения матричного умножения. Поэтому следует прямо указывать требуемый порядок матриц сомножителей.

Рассмотрим, например, матрицы $A^{3 \times 4}$ и $B^{4 \times 2}$. В этом случае можно умножить A на B , но нельзя умножить B на A . В результате допустимого умножения получится матрица C размерностью 3×2 . Пусть, скажем,

$$A^{3 \times 4} = \begin{pmatrix} 4 & 3 & 0 & 1 \\ 2 & 7 & 1 & 5 \\ 0 & 2 & 8 & 9 \end{pmatrix} \text{ и } B^{4 \times 2} = \begin{pmatrix} 2 & 1 \\ 0 & 3 \\ 4 & 7 \\ 3 & 9 \end{pmatrix}.$$

По определению умножения матриц нужно взять первую строку (4 3 0 1) матрицы A и первый столбец (2 0 4 3) матрицы B и, рассматривая их как вектора, вычислить скалярное произведение $4 \cdot 2 + 3 \cdot 0 + 0 \cdot 4 + 1 \cdot 3 = 11$, в результате получится элемент c_{11} матрицы-результата. Затем нужно взять вторую строку и, умножив скалярно на первый столбец, получить c_{21} , затем, умножив третью строку на первый столбец, получить c_{31} и т. д. Перебрав все строки и все столбцы, определим все элементы искомой матрицы:

$$C^{3 \times 2} = \begin{pmatrix} 4 \cdot 2 + 3 \cdot 0 + 0 \cdot 4 + 1 \cdot 3 & 4 \cdot 1 + 3 \cdot 3 + 0 \cdot 7 + 1 \cdot 9 \\ 2 \cdot 2 + 7 \cdot 0 + 1 \cdot 4 + 5 \cdot 3 & 2 \cdot 1 + 7 \cdot 3 + 1 \cdot 7 + 5 \cdot 9 \\ 0 \cdot 2 + 2 \cdot 0 + 8 \cdot 4 + 9 \cdot 3 & 0 \cdot 1 + 2 \cdot 3 + 8 \cdot 7 + 9 \cdot 9 \end{pmatrix} = \begin{pmatrix} 11 & 22 \\ 23 & 75 \\ 59 & 142 \end{pmatrix}.$$

В операции умножения могут участвовать как вектор-строки, так и вектор-столбцы, естественно, при выполнении условия, обеспечивающего возможность матричного умножения. Если количество элементов вектор-строки совпадает с количеством строк в матрице, то вектор-строку можно умножить (слева) на матрицу, в результате получится вектор-строка, количество компонент в которой совпадает с количеством столбцов матрицы. Пусть, например, имеется вектор $\vec{d} = (2, -3, 0, 1)$. Его можно рассматривать как вектор-строку $(2 \ -3 \ 0 \ 1)$ и умножить (слева) на любую матрицу, количество строк в которой равно 4. Скажем, для определенной ранее матрицы $B^{4 \times 2}$ получим

$$\vec{d}B = (2 \ -3 \ 0 \ 1) \begin{pmatrix} 2 & 1 \\ 0 & 3 \\ 4 & 7 \\ 3 & 9 \end{pmatrix} = (2 \cdot 2 - 3 \cdot 0 + 0 \cdot 4 + 1 \cdot 3 \quad 2 \cdot 1 - 3 \cdot 3 + 0 \cdot 7 + 1 \cdot 9) = (7 \ 2)$$

— вектор из двух элементов. В рассматриваемом примере умножить вектор-строку \vec{d} на матрицу B справа, то есть получить $B\vec{d}$, невозможно.

Аналогично, при умножении матрицы размерности $m \times n$ на вектор-столбец размерности $n \times 1$ получится вектор-столбец размерности $m \times 1$. Например, транспонируя вектор-строку \vec{d} , можно получить вектор-столбец \vec{d}^T и умножить на него матрицу A :

$$A\vec{d}^T = \begin{pmatrix} 4 & 3 & 0 & 1 \\ 2 & 7 & 1 & 5 \\ 0 & 2 & 8 & 9 \end{pmatrix} \begin{pmatrix} 2 \\ -3 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 4 \cdot 2 - 3 \cdot 3 + 0 \cdot 0 + 1 \cdot 1 \\ 2 \cdot 2 - 7 \cdot 3 + 1 \cdot 0 + 5 \cdot 1 \\ 0 \cdot 2 - 2 \cdot 3 + 8 \cdot 0 + 9 \cdot 1 \end{pmatrix} = \begin{pmatrix} 0 \\ -12 \\ -3 \end{pmatrix}.$$

Следует отметить важное свойство единичных матриц: умножение на единичную матрицу как справа, так и слева не изменяет матрицу-сомножитель: $AI = A$ и $IB = B$ (естественно, при соблюдении необходимого условия выполнения умножения). В этом смысле единичная матрица является аналогом единицы в поле действительных чисел. Возможность использования описанного математического аппарата для кодирования сообщений базируется на том, что любое двоичное слово, например $u = 1100_2$, можно рассматривать, трактовать как вектор $\vec{u} = (1, 1, 0, 0)$, эквивалентную вектор-строку $(1 \ 1 \ 0 \ 0)$ или вектор-столбец $\begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix}$. И наоборот, любые вектор, вектор-столбец и вектор-строку с двоичными компонентами можно считать двоичным словом. Поэтому двоичные слова, двоичные вектора, вектор-строки и вектор-столбцы рассматриваются как один и тот же объект, представленный в разных формах, удобных для использования в разных ситуациях.

С квадратными матрицами тесно связаны важные математические объекты — **определители** или **детерминанты**.

ВНИМАНИЕ! **Определителем** квадратной матрицы A называется число, полученное по определенным правилам из элементов матрицы. Для обозначения определителя матрицы A используются такие обозначения: $\det(A)$, $|A|$ или ΔA .

Самым простым является рекурсивное правило вычисления определителя. В соответствии с ним:

определитель матрицы первого порядка $A^{1 \times 1} = (a_{11})$ равен единственному ее элементу: $\det(A^{1 \times 1}) = a_{11}$;

для квадратной матрицы $A^{n \times n}$ порядка $n > 1$ определитель вычисляется с помощью разложения по первой строке:

$$\det(A^{n \times n}) = \sum_{i=1}^n (-1)^{1+i} a_{1i} M_{1i},$$

где M_{1i} — дополнительный минор элемента a_{1i} , равный определителю матрицы $A^{(n-1) \times (n-1)}$ порядка $n-1$, полученной из матрицы $A^{n \times n}$ вычеркиванием строки и столбца, на пересечении которых находится элемент a_{1i} .

После вычеркивания из матрицы n -го порядка любой одной строки и любого одного столбца, очевидно, получится матрица $(n-1)$ -го порядка, поэтому дополнительный минор по своей сути является определителем. Таким образом, это рекурсивное правило сводит вычисление определителя матрицы n -го порядка к вычислению n определителей матриц $(n-1)$ -го порядка — дополнительных миноров для каждого из элементов первой строки матрицы.

Применим это правило к определителю матрицы второго порядка:

$$\det(A^{2 \times 2}) = (-1)^{1+1} a_{11} M_{11} + (-1)^{1+2} a_{12} M_{12} = a_{11} M_{11} - a_{12} M_{12}.$$

Определим теперь дополнительные миноры M_{11} и M_{12} для каждого элемента первой строки. В соответствии с определением для получения минора M_{11} элемента a_{11} нужно

из матрицы $A^{2 \times 2} = \begin{pmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{pmatrix}$ вычеркнуть первую строку и первый столбец, в результате

получится матрица $\begin{pmatrix} a_{22} \end{pmatrix}$ первого порядка, определитель которой по определению равен a_{22} , следовательно, $M_{11} = a_{22}$. Аналогично, при определении минора M_{12} для элемента a_{12} из матрицы $A^{2 \times 2}$ нужно вычеркнуть первую строку и второй столбец, и получить $M_{12} = a_{21}$. Таким образом, находим, что определитель матрицы второго порядка вычисляется по формуле

$$\det(A^{2 \times 2}) = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11} a_{22} - a_{12} a_{21}.$$

Например,

$$\begin{vmatrix} 6 & 3 \\ 2 & 2 \end{vmatrix} = 6 \cdot 2 - 3 \cdot 2 = 6; \quad \begin{vmatrix} 1 & 2 \\ 2 & 4 \end{vmatrix} = 1 \cdot 4 - 2 \cdot 2 = 0.$$

Применение этого правила к расчету определителя третьего порядка сводит его вычисление к трем определителям второго порядка:

$$\det(A^{3 \times 3}) = \begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11} \begin{vmatrix} a_{22} & a_{23} \\ a_{32} & a_{33} \end{vmatrix} - a_{12} \begin{vmatrix} a_{21} & a_{23} \\ a_{31} & a_{33} \end{vmatrix} + a_{13} \begin{vmatrix} a_{21} & a_{22} \\ a_{31} & a_{32} \end{vmatrix},$$

которые вычисляются по полученным ранее простым формулам.

ВНИМАНИЕ! Вычислять значения определителя можно не только путем разложения по элементам первой строки — для этого можно использовать любую строку и любой столбец.

Определители играют важную роль в решении многих задач. В частности, справедливо следующее утверждение.

ВНИМАНИЕ! Система из n линейных алгебраических уравнений с n неизвестными имеет единственное решение, если определитель матрицы, составленной из коэффициентов левой части системы, отличен от нуля.

Это утверждение потребует при изучении так называемого БЧХ-кода.

Литература

1. *Аверьянов Г.П., Дмитриева В.В.* Современная информатика. — М.: НИЯУ МИФИ, 2011.
2. *Акулов В.А., Медведев Н.В.* Информатика. Базовый курс. — М.: Омега-Л, 2005.
3. *Аладьев В. З., Хунт Ю. Я., Шишаков М. Л.* Основы информатики. — М.: Филинь, 1999.
4. *Андерсон Д.* Дискретная математика и комбинаторика. — М.: Вильямс, 2003.
5. *Андреева Е., Фалина И.* Системы счисления и компьютерная арифметика. — М.: Лаборатория базовых знаний, 2000.
6. *Апокин И. А., Майстеров Л. Е.* История вычислительной техники. — М.: Наука, 1990.
7. *Ахо А., Сети Р., Ульман Дж.* Компиляторы: принципы, технологии, инструменты. — М.: Вильямс, 2001.
8. *Ахо А., Хопкрофт Дж., Ульман Дж.* Построение и анализ вычислительных алгоритмов. — М.: Мир, 1979.
9. *Ахо А., Хопкрофт Дж., Ульман Дж.* Структуры данных и алгоритмы. — М.: Вильямс, 2000.
10. *Ахо А., Ульман Дж.* Теория синтаксического анализа, перевода и компиляции. Т. 1–2. — М.: Мир, 1978.
11. *Бауэр Ф. Л., Гооз Г.* Информатика. — М.: Мир, 1990.
12. *Бейзер Б.* Тестирование черного ящика. Технологии функционального тестирования программного обеспечения и систем. — СПб.: Питер, 2004.
13. *Бен-Ари М.* Языки программирования. Практический сравнительный анализ. — М.: Мир, 2000.
14. *Беляев М.А., Лысенко В.В., Малинина Л.А.* Информатика — Ростов н/Д: Феникс, 2006.
15. *Берлекэмп Э.* Алгебраическая теория кодирования. — М.: Мир, 1971.
16. *Блейхуд Р.* Теория и практика кодов, контролирующих ошибки. — М.: Мир, 1988.
17. *Бозм Б. и др.* Характеристики качества программного обеспечения. — М.: Мир, 1991.
18. *Братко И.* Программирование на языке ПРОЛОГ для искусственного интеллекта. — М.: Мир, 1990.
19. *Братчиков И. Л.* Синтаксис языков программирования. — Новосибирск: Наука, 1977.
20. *Брауде Э. Дж.* Технологии разработки программного обеспечения. — СПб.: Питер, 2004.
21. *Брой М.* Информатика. Основополагающее введение. — М.: Диалог-МИФИ, 1996.

22. Брукс Ф. Мифический человеко-месяц, или Как создаются программные системы. — М.: Символ-Плюс, 2010.
23. Бутаков Е.А. Методы создания качественного программного обеспечения. — М.: Энергоатомиздат, 1984.
24. Буч Г. Объектно-ориентированный анализ и проектирование с примерами применения. — М.: Конкорд, 1992.
25. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя. — М.: ДМК Пресс, 2001.
26. Варшавский В. И., Мараховский В. Б., Песчанский В. А., Розенблюм Л. Я. Однородные структуры. — М.: Энергия, 1973.
27. Вельбицкий И. В., Ходаковский В. Н., Шолмов Л. И. Технологический комплекс производства программ на машинах ЕС ЭВМ и БЭСМ-6. — М.: Статистика, 1980.
28. Вендров А. М. CASE-технологии. Современные методы и средства проектирования информационных систем. — М.: Финансы и статистика, 1998.
29. Вернер М. Основы кодирования. — М.: Техносфера, 2004.
30. Вирт Н. Алгоритмы и структуры данных. — СПб.: Невский диалект, 2001.
31. Вирт Н., Гуткнехт Ю. Разработка ОС и компилятора. Проект Оберон (Классика программирования). — СПб.: БХВ-Петербург, 2012.
32. Воеводин В. В., Воеводин Вл. В. Параллельные вычисления. — СПб.: БХВ-Петербург, 2002.
33. Волкова В. Н., Денисов А. А. Теория систем и системный анализ. — М.: Юрайт, 2012.
34. Галлагер Р. Теория информации и надежная связь. — М.: Советское радио, 1974.
35. Гамма Э., Хелм Р., Джонсон Р., Влиссидес Дж. Приемы объектно-ориентированного программирования. Паттерны проектирования. — СПб.: Питер, 2001.
36. Гарднер М. Математические досуги. — М.: Мир, 1972.
37. Гашков Б. С. Системы счисления и их применение. — Библиотека «Математическое просвещение», 2004.
38. Гинзбург С. Математическая теория контекстно-свободных языков. — М.: Мир, 1970.
39. Гойвертс Я., Левитан С. Регулярные выражения. Сборник рецептов. — СПб.: Символ-Плюс, 2010.
40. Головешкин В. А., Ульянов М. В. Теория рекурсии для программистов. — М.: Физматлит, 2006.
41. Головкин Б. А. Параллельные вычислительные системы. — М.: Наука, 1980.
42. Горяев Ю.А. Информатика. — М.: МИЭМП, 2005.
43. Глушков В.Л. Основы безбумажной информатики. — М.: Наука, 1987.
44. Грехем Р., Кнут Д., Паташник О. Конкретная математика. Основание информатики. — М.: Мир, 1998.
45. Грошев А. С. Информатика. — Архангельск: ГТУ, 2010.
46. Грин Д., Кнут Д. Математические методы анализа алгоритмов. — М.: Мир, 1987.
47. Грис Д. Наука программирования. — М.: Мир, 1984.
48. Дал У., Дейкстра Э., Хоор К. Структурное программирование. — М.: Мир, 1975.

49. Дейкстра Э. Дисциплина программирования. — М.: Мир, 1978.
50. Душкин Р. В. Функциональное программирование на языке Haskell. — М.: ДМК Пресс, 2006.
51. Зрюмова А. Г., Зрюмов Е. А. Пронин С. П. Информатика. — Барнаул: АлтГТУ, 2011.
52. Зыков А. А. Основы теории графов. — М.: Вузовская книга, 2004.
53. Информатика / Под общ. ред. Данчула А.Н. М.: Изд-во РАГС, 2004.
54. Информатика. Базовый курс / Под ред. С. В. Симоновича. — СПб.: Питер, 2002.
55. Информатика / Под общ. ред. Хубаева Г. Н. — Ростов н/Д: Феникс, 2010.
56. Каймин В. А. Информатика. — М.: ИНФРА-М, 2001.
57. Касьянов В. Н. Лекции по теории формальных языков, автоматов и сложности вычислений. — Новосибирск: НГУ, 1995.
58. Карпов Ю. Г. Теория автоматов. — СПб.: Питер, 2003.
59. Карпов Ю. Г. Теория и технология программирования. Основы построения трансляторов. — СПб.: БХВ-Петербург, 2005.
60. Кватрани Т. Rational Rose 2000 и UML. Визуальное моделирование. — М.: ДМК Пресс, 2001.
61. Керниган Б., Пайк Р. Практика программирования. М.: Бином, 2001.
62. Керниган Б., Ритчи Д. Язык программирования Си. М.: Вильямс, 2007.
63. Классификация // Философский словарь. Под ред. И. Т. Фролова. — 4-е изд. — М.: Политиздат, 1981.
64. Ключко В. И. Кодирование информации: Курс лекций. — Краснодар: КубГТУ, 1998.
65. Кнут Д. Искусство программирования, том 1. Основные алгоритмы. М.: «Вильямс», 2006.
66. Колесник В. Д., Полтырев Г.Ш. Курс теории информации. — М.: Наука, 1982.
67. Колмогоров А. Н. Теория информации и теория алгоритмов. — М.: Наука, 1987.
68. Колмогоров А. Н. Алгоритм, информация, сложность. — М.: Знание, 1991.
69. Колмогоров А. Н., Успенский В. А. К определению алгоритма. УМН, 1958, Т. 13, № 4, с. 2–28.
70. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ. — М.: «Вильямс», 2006.
71. Королев Л. Н. Структуры ЭВМ и их математическое обеспечение. — М.: Наука, 1978.
72. Королев Л. Н., Миков А. И. Информатика. Введение в компьютерные науки. — М.: Высш. шк., 2003.
73. Криницкий Н. А. Алгоритмы вокруг нас. — М.: Наука, 1984.
74. Кричевский Р. Е. Сжатие и поиск информации. — М.: Наука, 1986.
75. Крупский В. Н. Введение в сложность вычислений. — М.: Факториал Пресс, 2006.
76. Крупский В. Н., Плиско В. Е. Математическая логика и теория алгоритмов. — М.: Academia, 2013. Вставлен 26.11.2016
77. Кудинов Ю. И., Пащенко Ф. Ф. Основы современной информатики. — М.: Лань, 2011.
78. Кудряшов Б. Д. Теория информации. — СПб.: Питер, 2009.

79. Куж С. Вычислительная сложность функций высшего типа. — М.: Мир, 1996.
80. Кулямин В. В. Методы верификации программного обеспечения. — М.: Институт Системного Программирования РАН, 2008.
81. Лавров С. С. Программирование. Математические основы, средства, теория. — СПб.: БХВ-Петербург, 2001.
82. Майерс Г., Баджетт Т., Сандлер К. Искусство тестирования программ. — М.: Диалектика, 2012.
83. Макарова Н. В., Волков В. Б. Информатика — СПб.: Питер, 2011.
84. Мак-Вильямс Ф., Дж. Ф., Слоэн Дж. А. Теория кодов, исправляющих ошибки. — М.: Связь, 1979.
85. Макконелл Дж. Основы современных алгоритмов. — М.: Техносфера, 2004.
86. Мальцев А. И. Алгоритмы и рекурсивные функции. — М.: Наука, 1986.
87. Марков А. А. О представлении рекурсивных функций. — Изв. АН СССР, сер. Мат. 1948, 13, № 5, с. 417–424.
88. Марков А. А. Теория алгоритмов. Тр. мат. ин-та АН СССР им. В. А. Стеклова., Т. 42, 1954.
89. Марков А. А., Нагорный Н. М. Теория алгорифмов. — М.: ФАЗИС, 1996.
90. Мельников С. В. Perl для профессиональных программистов. Регулярные выражения. — М.: Бином, 2007.
91. Мински М. Вычисления и автоматы. — М.: Мир, 1971.
92. Могилев А. В., Пак Н. И., Хеннер Е. К. Информатика. — М.: Academia, 2004.
93. Мозговой М. В. Классика программирования: алгоритмы, языки, автоматы, компиляторы. — СПб.: Наука и Техника, 2006.
94. Моисеев Н. Н. Математические методы системного анализа. — М.: Наука, 1984.
95. Молчанов А. Ю. Системное программное обеспечение. — СПб.: Питер, 2003.
96. Мышкис А. Д. Элементы теории математических моделей. — М.: КомКнига, 2007.
97. Наумов Л., Шалыто А. Клеточные автоматы. Реализация и эксперименты. // Мир ПК, 2003, № 8, С. 64-71.
98. Нейман Дж. Теория самовоспроизводящихся автоматов. — М.: Мир, 1971.
99. Носовский Г. В., Фоменко А. Т. Новая хронология Руси, Англии и Рима. — М.: Анвик, 1999 г.
100. Одинцов И. Профессиональное программирование. Системный подход. — СПб.: «БХВ-Петербург», 2002.
101. Павловская Т. А. C#. Программирование на языке высокого уровня. — СПб.: Питер, 2007.
102. Пайтген Х. О., Рихтер П. Х. Красота фракталов. Образы комплексных динамических систем. — М.: Мир, 1993.
103. Паронджанов В. Д. Язык ДРАКОН. Краткое описание. — М.: ДМК Пресс, 2009.
104. Питерсон У., Уэлдон Э. Коды, исправляющие ошибки. — М.: Мир, 1976.
105. Потанов В. Н. Теория информации. Кодирование дискретных вероятностных источников. — Новосибирск: НГУ, 1999.

106. *Пратт Т., Зелковиц М.* Языки программирования. Разработка и реализации. — СПб.: Питер, 2002.
107. *Пригожин И., Стенгерс И.* Порядок из хаоса. Новый диалог человека с природой. М.: Эдиториал УРСС, 2000.
108. Рекомендации по преподаванию информатики в университетах. Computing Curricula 2001: Computer Science. — СПб.: Изд-во СПбГУ, 2002.
109. *Разборов А. А.* Theoretical Computer Science: взгляд математика // Компьютера, № 2, 2001.
110. *Разборов А. А.* О сложности вычислений // Математическое просвещение. — МЦНМО, 1999. Том 3. Серия 3. С. 127–141.
111. *Роганов Е. А.* Основы информатики и программирования — М.: МГИУ, 2001.
112. *Савельев А. Я.* Основы информатики. — М.: Изд-во МГТУ им Н.Э. Баумана, 2001.
113. *Самарский А. А., Михайлов А. П.* Математическое моделирование. Идеи. Методы. Примеры. — М.: Физматлит, 2001.
114. *Свердлов С. З.* Языки программирования и методы трансляции. — СПб.: Питер, 2007.
115. *Себеста Р. У.* Основные концепции языков программирования. — М.: Вильямс, 2001.
116. *Симонович С. В.* Информатика. Базовый курс. 3-е изд. — СПб.: Питер, 2011.
117. *Синицын С. В., Налютин Н. Ю.* Верификация программного обеспечения. — М.: БИНОМ, 2008.
118. *Смирнов А. Д.* Архитектура вычислительных систем. — М.: Наука, 1990.
119. *Смит Б.* Методы и алгоритмы вычислений на строках. — М.: Вильямс, 2006.
120. *Соболь Б. В.* Информатика. — Ростов н/Д: Феникс, 2007.
121. *Советов Б. Я., Яковлев С. А.* Моделирование систем. — М.: Высш. шк., 2001.
122. *Соловьева Ф. И.* Введение в теорию кодирования. — Новосибирск: НГУ, 2006.
123. *Соммервел И.* Инженерия программного обеспечения. — М.: Изд. дом «Вильямс», 2002.
124. *Сотникова М. В.* Лекции по основам программирования. — СПб.: Изд. «Лемма», 2016.
125. *Стариченко Б. Е.* Теоретические основы информатики. — М.: Горячая линия–Телеком, 2003.
126. *Степанов А. Н.* Информатика. Шестое издание. — СПб.: Питер, 2009.
127. *Степанов А. Н.* Архитектура вычислительных систем и компьютерных сетей. — СПб.: Питер, 2007
128. *Стерлинг Л., Шапиро Э.* Искусство программирования на языке ПРОЛОГ. — М.: Мир, 1990.
129. *Стратонович Р. Л.* Теория информации. — М.: Сов. радио, 1975.
130. *Сухомлин В. А.* Принципы работы системы Эльбрус. — М.: Изд-во МГУ, 1981.
131. *Сухомлин В. А.* Введение в анализ информационных технологий. — М.: «Горячая линия — Телеком», 2003.
132. *Сэвидж Дж. Э.* Сложность вычислений. — М.: Факториал, 1998.

133. *Сэломон Д.* Сжатие данных, изображения и звука. — М.: Техносфера, 2004.
134. *Тарасов В. Б.* От многоагентных систем к интеллектуальным организациям: философия, психология, информатика. — М.: Эдиториал УРСС, 2002.
135. *Таганов Л. С., Пимонов А. Г.* Информатика. — Кемерово: КузбГТУ; 2010. с.
136. *Таненбаум Э.* Архитектура компьютера. — 6-е изд. — СПб.: Питер, 2015.
137. *Терехов А. В., Чернышев А. В., Чернышев В. Н.* Информатика. — Тамбов: ТГТУ, 2007.
138. *Терехов А. Н.* Технология программирования. — М.: БИНОМ, 2006.
139. *Тоффоли Т., Марголюс Н.* Машины клеточных автоматов. — М.: Мир, 1991.
140. *Успенский В. А.* Машина Поста. — М.: Наука, 1988.
141. *Фано Р. М.* Передача информации. Статистическая теория связи. — М.: Мир, 1965.
142. *Фаулер М., Скотт К.* UML в кратком изложении. Применение стандартного языка объектного моделирования. — М.: Мир, 1999.
143. *Фейнман Р.* Моделирование физики на компьютерах // Квантовый компьютер и квантовые вычисления. Ижевск: Регулярная и хаотическая динамика, 1999.
144. *Филд А., Харрисон П.* Функциональное программирование. — М.: Мир, 1993.
145. *Фицджеральд М.* Регулярные выражения. Основы. — М.: Вильямс, 2015.
146. *Фомин С. В.* Системы счисления — М.: Наука, 1987.
147. *Фридл Дж.* Регулярные выражения. — СПб.: Питер, 2008.
148. *Хартли Р.* Передача информации. Пер. с англ. в сб.: Теория информации и ее приложения. — М.: ИЛ, 1959.
149. *Хопкрофт Д., Мотвани Р., Ульман Дж.* Введение в теорию автоматов, языков и вычислений. — М.: Вильямс, 2002.
150. *Хювёнен Э., Сеппянен Й.* Мир Лиспа. Введение в язык Лисп и функциональное программирование. В 2-х томах — М.: Мир, 1990.
151. *Чисар И., Кернер Я.* Теория информации: Теоремы кодирования для дискретных систем без памяти. — М.: Мир, 1985.
152. *Шальто А., Туккель Н.* От тьюрингова программирования к автоматному // Мир ПК. 2002, №2.
153. *Шеннон К. Э.* Работы по теории информации и кибернетике. Пер. с англ. — М.: ИЛ, 1963.
154. *Шоломов Л. А.* Основы теории дискретных логических и вычислительных устройств. — М.: Наука, 1980.
155. Энциклопедия элементарной математики, Книга 1. Арифметика. Москва—Ленинград, Государственное издательство технико-теоретической литературы, 1951 г.
156. *Юров В.* Assembler. — 2-е изд. — СПб.: Питер, 2003.
157. *Яблонский С. В.* Введение в дискретную математику. — М.: Высшая школа, 2006.
158. *Яглом А. М., Яглом И. М.* Вероятность и информация. — М.: КомКнига, 2006.
159. *Claude E. Shannon.* A Mathematical Theory of Communication, Bell System Technical Journal, Vol. 27, pp. 379–423, 623–656, 1948.

160. Computer Science Curricula 2013. Curriculum Guidelines for Undergraduate Degree Programs in Computer Science. December 20, 2013. The Joint Task Force on Computing Curriculum Association for Computing Machinery (ACM) IEE Computer Society.
161. *Hoare C. A. R.* An axiomatic basis of computer programming. *Communication ACM*, v. 12 (October 1969), p. 576–580, 583.
162. *Naur P.* Proofs of algorithms by general snapshots. *Bit*, v. 6 (1969), p. 310–316.
163. *McCarthy J.* A basis for a mathematical theory of computation. *Proc. Western Joint Comp. Conf.*, Los Angeles, May 1961, p. 225–238.
164. *Emil L. Post.* Finite combinatory process — formulation 1//*The Journal of Symbolic Logic*. 1936, № 3.
165. *Floyd R.* Assigning meaning to programs. In *Mathematical Aspect of Computer Science*, Providence, 1967, p. 19–32.
166. *Wang Hao.* A variant to Turing’s theory of computing XII, XIII machines// *JACM*, 4, 1, 1957.
167. *Wolfram S. A.* *New Kind of Science*. — Wolfram Press, 2003.

Периодические издания

1. Ежемесячный журнал «Компьютер пресс», ISSN 0868-6157 (www.compress.ru).
2. Ежемесячный журнал «Мир ПК», ISSN 0235-3520 (www.pcworld.ru).
3. Ежемесячный журнал «Открытые системы. СУБД», ISSN 1028-7493 (www.osmag.ru).
4. Международный компьютерный еженедельник «Computerworld Россия», ISSN 1560-5213 (www.computerworld.ru).

Интернет

1. Ferra. Аналитические обзоры компьютеров и комплектующих, новости и цены компьютерного рынка // www.ferra.ru.
2. HardwarePortal. Обзоры и тесты компьютерного железа // www.hardwareportal.ru.
3. The collection of computer Science Bibliographies // iinwww.ira.uka.de/bibliography/index.html.
4. Whatis.ru. Все о компьютерах // www.whatis.ru.
5. Виртуальный компьютерный музей // www.computer_museum.ru.
6. Высокопроизводительные компьютеры // parallel.ru/computers.
7. Домашняя страница AMD // www.amd.com/ru-ru.
8. Железная столица. Новости аппаратных средств // www.stolica.ru.
9. Компьютерный музей // www.microsoft.com/museum.
10. Мир HARDware. Справочно-информационный сайт о компьютерной технике. Новости, обзоры, статьи об аппаратном и программном обеспечении компьютера. Тематические ссылки // www.hardw.com.ua.

11. Сайт компьютерных новостей. Техническая информация по аппаратному обеспечению: компьютерные новости, обзоры, советы и рекомендации и др. Ссылки на страницы производителей и поставщиков // www.ixbt.com.
12. Список 500 мощнейших суперкомпьютеров // www.top500.org/lists/plists.php?TV=1&M=11&Y=2004.
13. Тематический портал о компьютерах // hardvision.ru.

Краткая биографическая справка

1. **Аккерман Вильгельм Фридрих** (*нем.* Wilhelm Friedrich Ackermann; 1896–1962 гг.) — немецкий математик.
2. **Архимед** (*др.-греч.* Ἀρχιμήδης; 287–212 гг. до н.э.) — древнегреческий математик, физик и инженер из Сиракуз, сделал множество открытий в геометрии, заложил основы механики, гидростатики, автор ряда важных изобретений.
3. **Атанасов Джон Винсент** (*англ.* John Vincent Atanasoff; 1903–1995 гг.) — американский физик, математик и инженер-электрик болгарского происхождения, один из изобретателей первого электронного компьютера ABC.
4. **Ахо Альфред** (*англ.* Alfred Vaino Aho, 1941 г.р.) — канадский ученый в области информатики.
5. **Базилевский Юрий Яковлевич** (1912–1983 гг.) главный конструктор ЭВМ Стрела и оборонных систем СССР;
6. **Байрон Джордж Гордон** (*англ.* George Gordon Byron; 1788–1824 гг.) обычно именуемый просто лорд Байрон — британский поэт-романтик.
7. **Бэббидж Чарльз** (*англ.* Charles Babbage, 1791–1871 гг.) — английский математик, изобретатель первой аналитической вычислительной машины.
8. **Бекус Джон** (*англ.* John Backus, 1924–2007 гг.) американский специалист в области информатики, создатель первого алгоритмического языка высокого уровня Фортран и изобретатель нормальной формы метаязыка.
9. **Белоусов Борис Павлович** (1893–1970 гг.) — советский химик и биофизик.
10. **Бернулли Якоб** (*нем.* Jakob Bernoulli; 1655–1705 гг.) — швейцарский математик, один из основателей математического анализа и теории вероятностей.
11. **Берри Клиффорд Эдвард** (*англ.* Clifford Edward Berry; 1918–1963 гг.) помог Дж. Атанасову создать в 1939 году первый цифровой электронный ABC.
12. **Бессонов Николай Иванович** (1906–1963 гг.) — выдающийся русский советский инженер и ученый, разработчик теории синтеза логических схем, создатель релейной вычислительной машины РВМ 1.
13. **Бине Жак Филипп Мари** (*фр.* Jacques Philippe Marie Binet, 1786–1856 гг.) — французский математик, механик и астроном.
14. **Бодо Жан Морис Эмиль** (*фр.* Jean-Maurice-Émile Baudot,; 1845–1903 гг.) — французский инженер и изобретатель кода Бодо, способа кодирования символов для телетайпов. В честь Бодо названа единица символьной скорости — бод.
15. **Бонч-Бруевич Михаил Александрович** (1888–1940 гг.) — русский и советский радиотехник, член-корреспондент АН СССР, основатель отечественной радиоламповой

- промышленности, изобретатель триггера — базового элемента современных компьютеров. Под его руководством в СССР в 30-х годах XX века была создана самая мощная на тот момент в мире радиостанция «Коминтерн».
16. **Брайль Луи** (*фр.* Louis Braille; 1809–1852 гг.) — французский изобретатель шрифта для чтения с помощью осязания для лиц с нарушениями зрения.
 17. **Брук Исаак Семёнович** (1902–1974 гг.) — главный конструктор одной из первых советских ЭВМ М-1;
 18. **Буч Гради** (*англ.* Grady Booch; 1955 г.р.) — американский специалист в области программной инженерии, один из создателей унифицированного языка UML
 19. **Винер Норберт** (*англ.* Norbert Wiener; 1894–1964 гг.) — американский ученый, выдающийся математик и философ, основоположник кибернетики и теории искусственного интеллекта.
 20. **да Винчи Леонардо ди сер Пьеро** (*итал.* Leonardo di ser Piero da Vinci; 1452–1519 гг.) — итальянский художник, живописец, скульптор, архитектор, музыкант, писатель, изобретатель и ученый, один из крупнейших представителей эпохи Возрождения.
 21. **Вирт Никлаус** (*англ.* Niklaus Wirth, род. 15.02.1934 г.) — швейцарский ученый, один из известнейших теоретиков в области информатики, разработчик популярных во всем мире языков программирования Паскаль, Модула, Оберон.
 22. **Вольфрам Стивен** (*англ.* Stephen Wolfram, 1959 г.р.) — английский физик, математик, программист. Автор и ведущий разработчик системы компьютерной алгебры Mathematica.
 23. **Галуа Эварист** (*фр.* Évariste Galois; 1811–1832 гг.) выдающийся французский математик, один из основоположников теории групп.
 24. **Гамильтон Уильям Роуэн** (*англ.* William Rowan Hamilton, 1805–1865 гг.) — ирландский математик, механик и физик, создатель основ векторного анализа, вариационного исчисления, оптики.
 25. **Гаусс Иоганн Карл Фридрих** (*нем.* Johann Carl Friedrich Gauß; 1777–1825 гг.) — немецкий математик, механик, физик, астроном и геодезист
 26. **Гёдель Курт Фридрих** (*нем.* Kurt Friedrich Gödel; 1906–1978 гг.) — австрийский логик, математик, и философ математик, наиболее известный сформулированной и доказанной им теоремой о неполноте.
 27. **Гильберт Давид** (*нем.* David Hilbert; 1862–1943 гг.) — немецкий математик-универсал, внес значительный вклад в развитие многих областей математики.
 28. **Глушков Виктор Михайлович** (1923–1982 гг.) — академик АН СССР, советский математик и кибернетик, главный конструктор созданной в 1966 году первой в мире персональной ЭВМ МИР-1 (Машина для Инженерных Расчетов).
 29. **Горнер Уильям Джордж** (*англ.* William George Horner, 1786–1837 гг.) — английский математик.
 30. **Грей Франк** (*англ.* Frank Gray, 1887–1969 гг.) американский физик, исследователь лаборатории Bell, автор многочисленных изобретений в области телевидения и электротехники.
 31. **Грис Дэвид** (*англ.* David Gries, 1939 г.р.) — американский ученый в области информатики.

32. **Гутенберг Иоганн** (*нем.* Johannes Gutenberg; между 1397 и 1400–1468 гг.) — немецкий первопечатник, создал способ книгопечатания подвижными литерами.
33. **Дейкстра Эдсгер Вибе** (*англ.* Edsger Wybe Dijkstra 1930–2002 гг.) — нидерландский ученый, труды которого оказали существенное влияние на развитие информатики, один из разработчиков концепции структурного программирования и теории верификации.
34. **Дойл Конан Артур Игнейшус** (*англ.* Sir Arthur Ignatius Conan Doyle; 1859–1930 гг.) — английский писатель (по образованию врач), автор многочисленных приключенческих, исторических, публицистических, фантастических и юмористических произведений.
35. **Ершов Андрей Петрович** (1931–1988 гг.) — академик АН СССР, советский ученый, один из создателей теоретического и системного программирования, его работы оказали огромное влияние на формирование и развитие вычислительной техники в СССР и во всем мире.
36. **Жаботинский Анатолий Маркович** (1938–2008 гг.) — советский и американский биофизик, физикохимик.
37. **Жаккард Жозеф Мари** (*фр.* Joseph Marie Jacquard; 1752–1834 гг.) — французский изобретатель ткацкого станка для узорчатых материй, известного как машина Жаккарда.
38. **Заменгоф Лазарь Маркович** (*польск.* Ludwik Lejzer Zamenhof; 1859–1917 гг.) — польский врач-окулист и лингвист, известен как создатель эсперанто — наиболее успешного из сконструированных международных языков.
39. **Зив Яков** (*англ.* Jacob Ziv, 1931 г. р.) — израильский ученый-математик, специалист в области теории информации и теории кодирования.
40. **Кардано Джироламо** (*итал.* Gerolamo Cardano, 1501–1576 гг.) — итальянский математик и инженер, изобретатель карданова подвеса и карданного вала.
41. **Карцев Михаил Александрович** (1923–1983 гг.) — главный конструктор первых поколений советских ЭВМ для систем контроля космического пространства и предупреждения о ракетном нападении;
42. **Кахан Вильям Мортон** (*англ.* William Morton Kahan, р. 1933 г.) канадский ученый в области вычислительной математики и теории вычислительных систем.
43. **Керниган Брайан Уилсон** (*англ.* Brian Wilson Kernighan; 1942 г. р.) — канадский ученый в области компьютерных технологий, соавтор популярных руководств по языку Си.
44. **Клини (или Клейни) Стивен Коул** (*англ.* Stephen Cole Kleene; 1909–1994 гг.) — американский математик и логик.
45. **Колмогоров Андрей Николаевич** (1903–1987 гг.) — академик АН ССР, русский советский математик, один из крупнейших математиков XX века.
46. **Конвей Джон Хортон** (*англ.* John Horton Conway; 1937 г. р.) — английский математик, создатель клеточного автомата «Жизнь». Занимается теорией конечных групп, теорией чисел и т.д.
47. **Котельников Владимир Александрович** (1908–2005 гг.) — академик АН СССР, советский и российский ученый в области радиотехники, радиосвязи и радиолокации планет.

48. **Кох Нильс Фабиан Хельге** (*швед.* Niels Fabian Helge von Koch; 1870–1924 гг.) — шведский математик.
49. **Крамер Габриэль** (*фр.* Gabriel Cramer, 1704–1752 гг.) — швейцарский математик, один из создателей линейной алгебры.
50. **Кук Стивен Артур** (*англ.* Stephen Arthur Cook, 1939 г. р.) — американский ученый в области теории вычислительных систем.
51. **Кулмероэ Алан** (*англ.* Alain Colmerauer, 1941 г. р.) — французский ученый в области информатики, создатель языка программирования Prolog.
52. **Лавлейс Августа Ада Кинг**, (*англ.* Augusta Ada King Byron, Countess of Lovelace, 1815–1852 гг.) — английский математик, считается первым программистом в истории человечества.
53. **Лебедев Сергей Александрович** (1902–1977 гг.) — академик АН СССР, основоположник теории и практики создания вычислительной техники в СССР. Под его руководством начиная с 1947 года были созданы 15 типов советских ЭВМ. Сюда входит первая ламповая машина МЭСМ — малая электронно-счетная машина, семейство машин БЭСМ — большая электронно-счетная машина, семейство машин М и т.д., вплоть до семейства суперкомпьютеров Эльбрус.
54. **Леверье Урбен** (*фр.* Urbain Jean Joseph Le Verrier, 1811–1877 гг.) — французский математик, занимавшийся небесной механикой.
55. **Лейбниц Готфрид Вильгельм** (*нем.* Gottfried Wilhelm von Leibniz; 1646–1716 гг.) — саксонский философ, логик, математик, механик, физик, юрист, историк, дипломат, изобретатель и лингвист.
56. **Лемпель Абрахам** (*англ.* Abraham Lempel, 1936 г. р.) — израильский специалист в области информационных технологий, один из создателей семейства сжимающих кодов без потери информации.
57. **Леонардо Пизанский** (*ит.* Leonardo Pisano, 1170–1250 гг.) более известный под именем **Фибоначчи** — первый крупный математик средневековой Европы
58. **Люка Франсуа Эдуард Анатоль** (*фр.* François Édouard Anatole Lucas, 1842–1891 гг.) — французский математик, специалист по теории чисел.
59. **Маккарти Джон** (*англ.* John McCarthy, 1927–2011 гг.) — американский ученый в области информатики, основоположник функционального программирования, изобретатель языка Лисп.
60. **Максвелл Джеймс Клерк** (*англ.* James Clerk Maxwell 1831–1879 гг.) — английский физик и математик, основатель классической электродинамики.
61. **Марков Андрей Андреевич** (1856–1922 гг.) — академик, русский математик, внесший большой вклад в математический анализ, теорию вероятностей и теорию чисел.
62. **Марков Андрей Андреевич** (1903–1979 гг.) — советский математик, сын известного русского математика А.А. Маркова (старшего), основоположник советской школы конструктивной математики.
63. **Матиясевич Юрий Владимирович** (1947 г. р.) — советский и российский математик, академик РАН.
64. **Менделеев Дмитрий Иванович** (1834–1907 гг.) — великий русский ученый-энциклопедист, химик, физик,

65. **Мински Марвин Ли** (*англ.* Marvin Lee Minsky; 1927–2016 гг.) — известный американский ученый в области искусственного интеллекта
66. **Морзе Сэмюэл Финли Бриз** (*англ.* Samuel Finley Brees Morse, 1791–1872 гг.) — американский изобретатель и художник.
67. **Мочли Джон Уильям** (*англ.* John William Mauchly; 1907–1980 гг.) — американский физик, инженер. Один из разработчиков первого в мире электронного компьютера ENIAC.
68. **Муавр Абрахам** (*фр.* Abraham de Moivre, 1667–1754 гг.) — английский математик французского происхождения
69. **Найквист Гарри** (*англ.* Harry Nyquist; 1889–1976 гг.) — один из основателей теории информации.
70. **Наумов Борис Николаевич** (1927–1988 гг.) — академик АН СССР, советский ученый в области автоматизации.
71. **Наур Питер** (*дат.* Peter Naur, род. 1928 г.) — датский специалист в области информатики, один из разработчиков знаменитого в свое время алгоритмического языка Алгол-60, уточнил нормальные формы Бекуса.
72. **Нейман Джон** (*англ.* John von Neumann; 1903–1957 гг.) — венгро-американский математик, внесший важный вклад в квантовую физику, квантовую логику, функциональный анализ, теорию множеств, информатику, экономику и другие отрасли науки.
73. **Однер Вильголд Теофил** (*швед.* Willgodt Theophil Odhner, 1845–1905 гг.) — шведско-русский механик, изобретатель, разработчик успешной конструкции арифмометра.
74. **Паскаль Блез** (*фр.* Blaise Pascal; 1623–1662 гг.) — французский математик, механик, физик, литератор и философ. Классик французской литературы, один из основателей математического анализа, теории вероятностей и проективной геометрии, создатель первых образцов счетной техники, автор основного закона гидростатики.
75. **Пенроуз Роджер** (*англ.* Roger Penrose, 1931 г. р.) — английский физик и математик, автор теории твисторов (квантовая физика)
76. **Пост Эмиль Леон** (*англ.* Post Emil Leo; 1897–1954 гг.) — американский математик и логик, предложил абстрактную вычислительную машину, которую назвали в его честь.
77. **Рамеев Башир Искандарович** (1918–1994 гг.) — советский ученый-изобретатель, разработчик первых советских ЭВМ Стрела и Урал 1.
78. **Рембо Джеймс** (*англ.* James Rumbaugh; 1947 г. р.) — американский ученый в области информатики, один из разработчиков технологии объектного моделирования и языка UML.
79. **Ритчи Деннис Макалистэйр** (*англ.* Dennis MacAlistair Ritchi; 1941–2011 гг.) — компьютерный специалист, создатель языка программирования Си, принимал участие в разработке операционной системы Unix.
80. **Стамма Филипп** (*англ.* Philipp Stamma; 1705–1755 гг.) — арабский шахматист и шахматный композитор XVIII века.
81. **Серпинский Вацлав Франциск** (*польск.* Waclaw Franciszek Sierpiński; 1882–1969 гг.) — польский математик, известен трудами по теории множеств, теории чисел, топологии.

82. **Стирлинг Джеймс** (*англ.* James Stirling, 1692–1770 гг.) — шотландский математик.
83. **Тейлор Брук** (*англ.* Brook Taylor; 1685–1731 гг.) — английский математик, именем которого называется известная формула, выражающая значение функции через значения всех ее производных в одной точке.
84. **Тьюринг Метисон Алан** (*англ.* Alan Mathison Turing; 1912–1954 гг.) — английский математик, логик, криптограф, оказавший существенное влияние на развитие информатики.
85. **Уайлс Эндрю Джон** (*англ.* Sir Andrew John Wiles; 1953 г. р.) — английский и американский математик.
86. **Уилкс Морис Винсент** (*англ.* Maurice Vincent Wilkes; 1913–2010 гг.) — британский ученый в области компьютерных наук.
87. **Улам Станислав Мартин** (*польск.* Stanisław Marcin Ulam; 1909–1984 гг.) — польский и американский математик.
88. **Ульман Джеффри Дэвид** (*англ.* Jeffrey David Ullman, 1942 г. р.) — американский исследователь в области информатики.
89. **Успенский Владимир Андреевич** (1930 г. р.) — советский и российский математик, лингвист, публицист.
90. **Уэлч Терри** (*англ.* Terry Arthur Welch, 1949 г. р.) — американский специалист в области информационных технологий.
91. **Фано Роберт Марио** (*ит.* Robert Mario Fano; 1917–2016 гг.) — италяно-американский ученый, известный своими работами по теории информации.
92. **Фидий** (*др.-греч.* Φειδίας, ок. 490 до н. э. — ок. 430 до н. э.) — древнегреческий скульптор и архитектор, один из величайших художников периода высокой классики.
93. **Флеминг Джон Амброс** (*англ.* John Ambrose Fleming; 1849–1945 гг.) — английский ученый в области радиотехники и электротехники.
94. **Флойд Роберт** (*англ.* Robert W Floyd, 1936–2001 гг.) — американский ученый в области информатики и теории вычислительных систем.
95. **Фоменко Анатолий Тимофеевич** (1945 г. р.) — академик РАН, крупный советский и российский математик.
96. **Фурье Жан Батист Жозеф** (*фр.* Jean Baptiste Joseph Fourier; 1768–1830 гг.) — французский математик и физик.
97. **Хао Ван** (*кит.* Wang Hao, 1917–1995 гг.) — китайско-американский математик, философ.
98. **Хартли Ральф Винтон Лайон** (*англ.* Ralph Vinton Lyon Hartley; 1888–1970 гг.) — американский ученый в области электроники, предложил логарифмическую меру количества информации.
99. **Хаффман Дэвид Алберт** (*англ.* David Albert Huffman; 1925–1999 гг.) — американский ученый в области теории информации.
100. **Хемминг Ричард Уэсли** (*англ.* Richard Wesley Hamming; 1915–1998 гг.) — американский математик, работы которого в сфере теории информации оказали существенное влияние на компьютерные науки и телекоммуникации.
101. **Хоар Чарльз Энтони Ричард** (*англ.* Charles Antony Richard Hoare, 1934 г. р.) — английский ученый в области информатики.

102. **Хопкрофт Джон Эдвард** (*англ.* John Edward Hopcroft, 1939 г. р.) — американский ученый в области информатики
103. **Холлерит Герман** (*англ.* Herman Hollerith, 1860–1929 гг.) — американский инженер, известен как создатель электрической табулирующей системы.
104. **Хомский Аврам Ноам** (*англ.* Avram Noam Chomsky; 1928 г. р.) — американский лингвист, автор классификации формальных языков.
105. **аль-Хорезми Абу Абдулах Мухаммед ибн Муса** (около 783–850 гг.) — математик, астроном и географ, основатель классической алгебры.
106. **Шеннон Клауд Элвуд** (*англ.* Claude Elwood Shannon; 1916–2001 гг.) — американский инженер и математик, основатель теории информации.
107. **Шиккард Вильгельм** (*нем.* Wilhelm Schickard; 1623–1635 гг.) — немецкий ученый, астроном, математик и востоковед, создатель первого арифмометра.
108. **Щерба Лев Владимирович** (1880–1944 гг.) — российский и советский лингвист, академик АН СССР, внесший большой вклад в развитие психолингвистики, лексикографии и фонологии.
109. **Цузе Конрад** (*нем.* Konrad Zuse; 1910–1995 гг.) — немецкий инженер, известен как создатель первого работающего компьютера и первого языка программирования высокого уровня.
110. **Чебышев Пафнутий Львович** (1821–1894 гг.) — русский математик и механик, основоположник петербургской математической школы, академик Петербургской академии наук и еще 24 академий мира.
111. **Чен Питер Пин-Шен** (*англ.* Peter Pin-Shen Chen; 1947 г. р.) профессор компьютерных наук в университете штата Луизиана (США), предложил модель «сущность-связь» и систему обозначений для ER-диаграмм.
112. **Чёрч Алонзо** (*англ.* Alonzo Church; 1903–1995 гг.) — американский математик и логик, внесший значительный вклад в основы информатики.
113. **Эйкен Говард Хатауэй** (*англ.* Howard Hathaway Aiken; 1900–1973 гг.) — американский инженер, руководил работами по созданию первого американского компьютера Марк 1.
114. **Эккерт Джон Проспер** (*англ.* Eckert John Prosper; 1919–1995 гг.) — американский инженер. Один из разработчиков первого в мире электронного компьютера ENIAC.
115. **Якобсон Ивар Ялмар** (*швед.* Ivar Hjalmar Jacobson; 1939 г. р.) — шведский ученый в области информатики, внесший вклад в развитие языка UML.