

Приложение Tip Calculator App

4

Создание приложения Android с помощью Java

В этой главе...

- Создание графического интерфейса пользователя с помощью макета `TableLayout`.
- Использование окна `Outline` модуля `ADT Plugin` в среде `Eclipse` для добавления компонентов GUI в макет `TableLayout`.
- Непосредственное редактирование кода XML макета GUI для настройки свойств, которые могут быть недоступны в `Visual Layout Editor` и в окне `Properties` для `Eclipse`.
- Использование компонентов GUI `TextView`, `EditText` и `SeekBar`.
- Применение свойств объектно-ориентированного программирования на Java, включая классы, анонимные внутренние классы, объекты, интерфейсы и наследование, применяемые для создания приложения Android.
- Изменение отображаемого текста путем программного взаимодействия с компонентами GUI.
- Использование обработки событий при взаимодействии с пользователем с помощью компонентов `EditText` и `SeekBar`.

4.1. Введение

Приложение `Tip Calculator` (рис. 4.1) вычисляет и отображает чаевые на основании суммы счета в ресторане. После ввода пользователем суммы счета приложение вычисляет и отображает размер чаевых и величину итогового счета (с учетом чаевых). При этом используются три процентных ставки для расчета чаевых: 10 %, 15 % и 20 %. Пользователь может указать собственную ставку, используемую при расчете чаевых. Для этого

нужно переместить ползунок компонента Seekbar, после чего обновляется величина процентной ставки, которая отображается в правой части компонента Seekbar. Мы выбрали величину 18 % в качестве заданной по умолчанию пользовательской процентной ставки для чаевых. Это связано с тем, что многие рестораны используют эту процентную ставку для компаний, состоящих из шести человек и более. Предполагаемые величины чаевых и общего счета могут быть изменены пользователем.

ПРИМЕЧАНИЕ

Цифровая клавиатура, показанная на рис. 4.1, может отличаться от вашей в зависимости от используемого устройства AVD либо версии Android, установленной на устройстве.

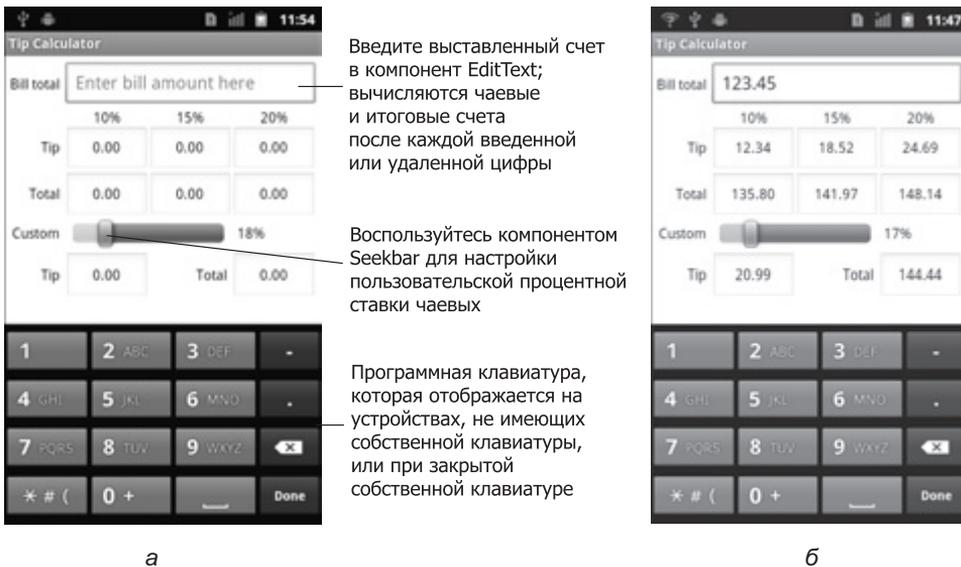


Рис. 4.1. Ввод суммы счета и подсчет чаевых: а — начальный интерфейс пользователя, появляющийся после касания пользователем компонента EditText Bill total и отображения цифровой клавиатуры; б — графический интерфейс после ввода выставленного счета на \$123,45 и изменения процентной ставки чаевых на 17 %

Начнем с тестирования приложения, заключающееся в вычислении стандартной и заданной пользователем ставок чаевых. Затем вас ожидает обзор технологий, применяемых для создания приложения. После этого мы создадим графический интерфейс пользователя (GUI) с помощью окна Outline в Eclipse. Это окно будет использоваться для добавления компонентов GUI. В графическом редакторе Visual Layout Editor мы увидим, как выглядит будущий GUI. Большая часть XML-кода для этого GUI будет сгенерирована с помощью инструментов плагина ADT. Потом мы отредактируем XML-код, что позволит настроить свойства, недоступные в окне Properties. И напоследок рассмотрим окончательный код приложения и проведем его подробный анализ.

4.2. Тестирование приложения Tip Calculator

Откройте и выполните приложение

Откройте интегрированную среду разработки Eclipse, импортируйте проект Tip Calculator и выполните следующие действия:

1. *Откройте диалоговое окно Import (Импорт)*. Чтобы открыть диалоговое окно Import, выполните команды File ▶ Import... (Файл ▶ Импорт...).
2. *Импортируйте проект приложения Tip Calculator*. В диалоговом окне Import раскройте узел General (Общие), выберите параметр Existing Projects into Workspace (Существующие проекты в рабочую область), а потом щелкните на кнопке Next> (Далее>) для выполнения шага Import Projects (Импорт проектов). Установите флажок Select root directory (Выбрать корневой каталог) и щелкните на кнопке Browse... (Просмотр...). В диалоговом окне Browse For Folder (Просмотр папок) найдите папку TipCalculator (в папке примеров книги), выделите ее и щелкните на кнопке OK. Щелкните на кнопке Finish (Готово) для выполнения импорта проекта в среду Eclipse. Проект появится в окне Package Explorer, находящемся в левой части окна Eclipse.
3. *Запустите приложение Tip Calculator*. В окне среды Eclipse щелкните правой кнопкой мыши на проекте TipCalculator в окне Package Explorer, затем в появившемся меню выберите команды Run As ▶ Android Application (Выполнить как ▶ Приложение Android). В результате приложение Tip Calculator начнет выполняться на виртуальном устройстве AVD, которое было создано так, как описано в разделе «Подготовительные действия» из вводной части книги.

ПРИМЕЧАНИЕ

Если было создано несколько устройств AVD либо к компьютеру подключено какое-либо физическое устройство Android, выберите одно из них для выполнения приложения.

Ввод выставленного счета

Коснитесь компонента EditText, который называется Bill Total, чтобы отобразить числовую клавиатуру. С помощью этой клавиатуры введите число 123,45.

ПРИМЕЧАНИЕ

Если нажатие клавиш приводит к появлению японских иероглифов, коснитесь пальцем элемента EditText под названием Bill Total и удерживайте палец пару секунд. В появившемся списке параметров выберите параметр Input method (Метод ввода). Потом выберите клавиатуру Android из второго списка параметров.

Если при вводе допущена ошибка, нажмите клавишу удаления, чтобы удалить последнюю введенную цифру. Компоненты EditText, имеющие заголовки 10 %, 15 % и 20 %, отображают размеры чаевых и общий счет для заранее заданных процентов чаевых (рис. 4.1, б). Также на экране отображаются компоненты EditText для пользовательских чаевых, общей суммы чаевых и общего счета для заданного по умолчанию значения 18 %, используемого при вычислении пользовательских чаевых. Все компоненты

`EditText`, используемые для вычисления чаевых и счета, обновляются после каждого ввода или удаления числа.

Выбор процентной ставки для пользовательских чаевых

Воспользуйтесь компонентом `SeekBar` для ввода процентной ставки, используемой при вычислении пользовательских чаевых. Перетаскивайте ползунок `SeekBar` до тех пор, пока процентная ставка для пользовательских чаевых не будет равна 17 %. Сумма чаевых и общего счета для выбранной процентной ставки появится в компонентах `EditText`, находящихся ниже компонента `SeekBar`. По умолчанию компонент `SeekBar` позволяет выбирать значения, находящиеся в диапазоне от 0 до 100.

4.3. Обзор применяемых технологий

В этой главе будут использоваться многие объектно-ориентированные свойства Java, включая классы, анонимные внутренние классы, объекты, методы, интерфейсы и наследование. Мы создадим подкласс класса `Android Activity`, с помощью которого определяется, что произойдет в случае запуска приложения на выполнение, а также логика приложения `Tip Calculator`. Будет реализовано программное взаимодействие с компонентами `EditText`, `TextView` и `SeekBar`. Эти компоненты создаются с помощью `Visual Layout Editor` и окна `Outline` в `Eclipse`, а также ручного редактирования XML-разметки, используемой для создания GUI. Компонент `EditText`, который часто называется текстовым полем или текстовым окном в других GUI-технологиях, — это подкласс компонента `TextView` (представлен в главе 3). Компонент `EditText` отображает текст либо принимает текст, вводимый пользователем. Компонент `SeekBar`, в других GUI-технологиях называемый ползунком, представляет целое число, заданное по умолчанию в диапазоне 0–100, которое может быть выбрано пользователем. Для организации взаимодействия между пользователем и интерфейсом пользователя применяется обработка событий и анонимные внутренние классы.

4.4. Создание графического интерфейса приложения

В этом разделе создается графический интерфейс приложения `Tip Calculator` с помощью инструментов модуля `ADT Plugin`. В конце этого раздела представлен код XML, сгенерированный модулем `ADT Plugin` для макета этого приложения. Мы подробно рассмотрим шаги, используемые для создания графического интерфейса. В следующих главах книги будут описаны новые свойства графического интерфейса каждого приложения, а также представлена завершенная XML-разметка интерфейса приложения с выделенными измененными фрагментами.

ПРИМЕЧАНИЕ

После выполнения операций, описанных в этом разделе, будет сформирован графический интерфейс, который будет отличаться от интерфейса на рис. 4.1 до тех пор, пока не будут выполнены шаги, описанные в разделах 4.4.2–4.4.4.

4.4.1. Знакомство с классом `TableLayout`

В приложении, создаваемом в этой главе, используется объект `TableLayout` (см. рис. 4.2) для расположения компонентов GUI в шести строках и четырех столбцах. Каждая ячейка, определенная объектом `TableLayout`, может быть пустой или включать один компонент, в качестве которого может выступать макет, *содержащий* другие компоненты. В строках 0 и 4 на рис. 4.2 показан компонент, который может *охватывать* несколько столбцов. Для создания строк используются объекты `TableRow`. Количество столбцов в `TableLayout` определяется с помощью объекта `TableRow`, который содержит большинство компонентов. Высота каждой строки определяется наиболее длинным компонентом в этой строке (обратите внимание на рис. 4.2, где строки 1 и 4 короче других строк). Аналогично ширина столбца определяется наиболее широким элементом этого столбца (если не разрешить «растягивание» столбцов таблицы для заполнения экрана по ширине, в результате чего столбцы становятся шире). По умолчанию компоненты добавляются в строку в направлении слева направо. Можно указать точное положение компонента с помощью нумерации, которая по умолчанию начинается с 0.

Дополнительные сведения о классе `TableLayout` можно найти на веб-сайте: developer.android.com/reference/android/widget/TableLayout.html, а сведения о классе `TableRow` — на веб-сайте: developer.android.com/reference/android/widget/TableRow.html.

Строки и столбцы объекта `TableLayout`

	Столбец 0	Столбец 1	Столбец 2	Столбец 3
Строка 0	Bill total	0.00		
Строка 1		10%	15%	20%
Строка 2	Tip	0.00	0.00	0.00
Строка 3	Total	0.00	0.00	0.00
Строка 4	Custom			18%
Строка 5	Tip	0.00	Total	0.00

Рис. 4.2. Класс `TableLayout` с подписанными строками и столбцами, используемый для создания GUI приложения Tip Calculator

На рис. 4.3 показаны имена всех компонентов GUI, применяемых для формирования графического интерфейса приложения. Используемое соглашение о наименовании предусматривает использование имени класса компонента GUI в качестве значения свойства `Id` каждого компонента в XML-разметке и в имени переменной каждого компонента в коде Java.

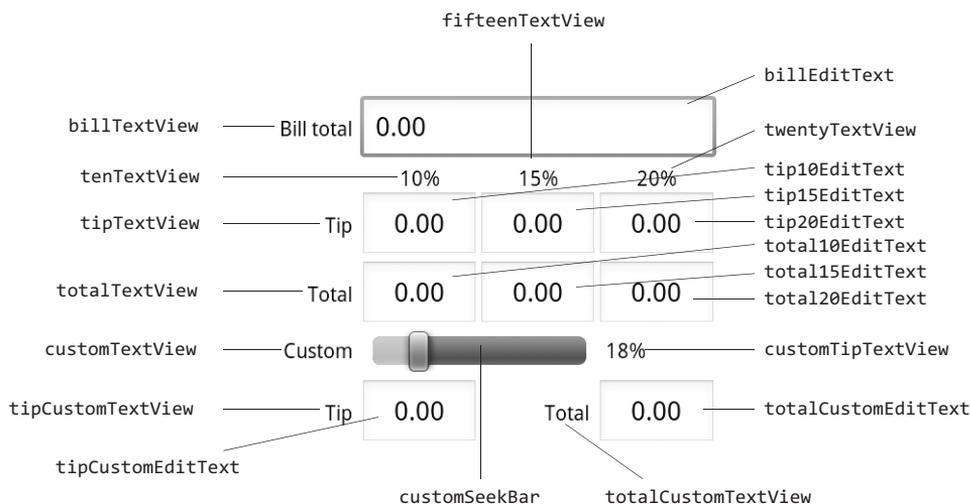


Рис. 4.3. Компоненты GUI приложения Tip Calculator, в качестве названий которых используются значения свойства Id

4.4.2. Создание проекта, добавление класса `TableLayout` и компонентов

А теперь приступим к созданию GUI, показанного на рис. 4.2. Начнем с формирования базового макета и элементов управления, потом настроим свойства элементов управления, завершив тем самым проект. После добавления компонентов в каждую строку объекта `TableLayout` можно настроить свойства `Id` и `Text` этих компонентов (см. рис. 4.3). В разделе 3.5 предыдущей главы упоминалось о том, что при разработке проекта значения литеральных строк должны включаться в файл `strings.xml`, который находится в папке `res/values` приложения. Эта методика особенно полезна в тех случаях, когда вы намереваетесь локализовать приложение для его использования на разных языках. Для компонентов `TextView` с названиями 10 %, 15 % и 20 % строковые ресурсы не используются. Убедитесь в том, что выполнили шаги, требуемые для создания GUI, компоненты которого расположены в заданном порядке. Если заданный порядок нарушен, измените расположение компонентов в окне `Outline` либо в файле `main.xml`.

При выполнении следующих шагов с помощью окна `Outline` выполняется добавление компонентов в соответствующие объекты `TableRow` из класса `TableLayout`. При работе с более сложными структурами, например `TableLayouts`, с помощью `Visual Layout Editor` довольно трудно просматривать сложную многоуровневую структуру макета и помещать компоненты на нужный уровень. В окне `Outline` отображается разветвленная структура GUI, в связи с чем задача проектирования существенно упрощается. Например, можно выбрать соответствующую строку в `TableLayout` и добавить в эту строку выбранный вами компонент GUI.

Шаг 1. Создание проекта TipCalculator

В среде Eclipse допускается создание лишь одного проекта с заданным именем для одной рабочей области. Поэтому перед выполнением этого шага удалите из рабочей области существующее приложение Tip Calculator, которое ранее уже тестировалось. Для выполнения этого действия щелкните на названии приложения правой кнопкой мыши и в контекстном меню выберите параметр Delete (Удалить). В появившемся диалоговом окне отмените установку флажка Delete project contents on disk is not selected (Удалить содержимое проекта на диске) и щелкните на кнопке OK. В результате произойдет удаление проекта из рабочей области, а папка проекта, находящаяся на диске, останется нетронутой. Затем создайте новый проект Android под названием TipCalculator. В диалоговом окне New Android Project (Новый проект Android) укажите следующие значения и щелкните на кнопке Finish (Готово):

- Build Target (Операционная система): выберите Android 2.3.3.
- Application name (Имя приложения): Tip Calculator.
- Package name (Название пакета): com.deitel.tipcalculator.
- Create Activity (Создать деятельность): TipCalculator.
- Min SDK Version (Минимальная версия SDK): 10. (*Примечание.* Эта версия SDK соответствует версии Android 2.3.3, хотя в этом приложении не будут использоваться функции, присущие именно Android 2.3.3. Если планируется выполнять это приложение на виртуальном устройстве Android (AVD) либо на реальном устройстве, на котором выполняется более ранняя версия Android, параметру Min SDK присвойте меньшее значение. Например, если выбрано значение 8 для этого параметра, приложение будет выполняться на версии Android 2.2 или более старшей.)

Шаг 2. Удаление и повторное создание файла main.xml

Для создаваемого в этой главе приложения нужно заменить стандартный файл main.xml другим, использующим TableLayout, в котором компоненты расположены друг относительно друга. Чтобы заменить стандартный файл main.xml, выполните следующие действия:

1. Чтобы удалить файл main.xml, щелкните на нем (в папке проектов /res/layout) и в контекстном меню выберите параметр Delete (Удалить).
2. Правой кнопкой мыши щелкните на папке layout и выберите команды New ▶ Other... (Создать ▶ Другое...). Отобразится диалоговое окно New (Создать).
3. В узле Android выберите параметр Android XML File (XML-файл Android). После щелчка на кнопке Next> (Далее>) появится диалоговое окно New Android XML File (Создать XML-файл Android).
4. Выберите название файла main.xml и выделите TableLayout, потом щелкните на кнопке Finish.

Шаг 3. Конфигурирование Visual Layout Editor для использования подходящей библиотеки Android SDK

После выполнения предыдущего шага в окне Visual Layout Editor открывается новый файл main.xml. Если установлено несколько библиотек Android SDK, модуль ADT

Plugin выбирает последнюю библиотеку в качестве стандартной, используемой для создания интерфейса (на вкладке Graphical Layout), причем этот выбор независим от SDK, выбранной при создании проекта. В раскрывающемся списке селектора SDK, находящемся в правой верхней части вкладки Graphical Layout, выберите параметр Android 2.3.3 (см. рис. 3.7). В результате разрабатываемый графический интерфейс будет совместим с устройствами Android 2.3.3.

Шаг 4. Настройка размеров и разрешения окна Visual Layout Editor

В раскрывающемся списке Device Configurations (Конфигурации устройств), находящемся в левой верхней части вкладки Graphical Layout (см. рис. 3.11), выберите параметр 3.7in WVGA (Nexus One). В результате будет настроена область конфигурирования для устройств с экраном, имеющим разрешение 480×800 (WVGA).

Шаг 5. Конфигурирование параметров объекта TableLayout

В окне Outline выберите параметр TableLayout. На экране появится окно Properties, в котором представлены следующие свойства.

- Background (Фон): #FFF.
- Id (Идентификатор): @+id/tableLayout.t.
- Padding (Отступ): 5 dp.
- Stretch columns (Растяжение столбцов): 1, 2, 3.

По умолчанию параметрам Layout width и Layout height присваивается значение match_parent, в результате чего макет занимает весь экран. Параметру Padding присваивается значение 5 dp, в результате чего вдоль границы всего макета создается кайма шириной в 5 независимых от плотности пикселей. С помощью параметра Stretch columns (представлен в XML атрибуте android:stretchColumns, см. листинг 4.1, строка 8) определяется растяжение по горизонтали столбцов 1–3, чтобы заполнить макет по ширине. Ширина столбца 0 соответствует ширине наиболее широкого содержащегося в нем элемента вместе с отступом вокруг этого элемента.

Шаг 6. Добавление компонентов TableRow

На этом шаге с помощью окна Outline добавим шесть компонентов TableRow в объект TableLayout. Выполните следующие действия:

1. В окне Outline щелкните правой кнопкой мыши на компоненте tableLayout и выберите добавляемый компонент Outline.
2. Повторите этот процесс еще пять раз.

При каждом повторении процесса не забудьте щелкнуть правой кнопкой мыши на tableLayout с тем, чтобы TableRow корректно вкладывались в TableLayout. Свойству Id для каждого TableRow автоматически присваиваются значения от tableRow1 до tableRow6 соответственно. Поскольку нумерация столбцов начинается с 0, соответствующим образом были изменены свойства Id для TableRow, которые получили значения от tableRow0 до tableRow5 соответственно. Также выберите каждый объект TableRow и присвойте его свойству Layout width значение match_parent. В результате строки займут все пространство макета. Чтобы выполнить эту операцию для всех компонентов TableRow

одновременно, щелкните на первом компоненте `TableRow`, отображенном в окне `Outline`. Затем, удерживая клавишу `Shift`, щелкните на последнем компоненте `TableRow` в окне `Outline`, чтобы выделить все шесть компонентов. Затем присвойте значения свойствам.

Шаг 7. Добавление компонентов в `tableRow0`

На этом шаге компоненты `TextView` и `EditText` будут добавлены в `tableRow0`. Выполните следующие действия:

1. Перетащите компонент `TextView` (`billTextView`) из раздела `Form Widgets` (Виджеты форм) палитры в `tableRow0`, находящийся в окне `Outline`.
2. Перетащите компонент `EditText` (`billEditText`) из раздела `Form Widgets` (Виджеты форм) палитры в `tableRow0`, находящийся в окне `Outline`.
3. Присвойте значения свойствам `Id` и `Text` для каждого компонента. Чтобы получить быстрый доступ к этим свойствам, щелкните правой кнопкой мыши на компоненте в окне `Outline` и в контекстном меню выберите команды `Edit ID...` (Изменить идентификатор...) и `Edit Text...` (Изменить текст...) соответственно.

Важно перетащить указанные выше компоненты на подходящие объекты `TableRow` в окне `Outline`, чтобы они были вложены в требуемый объект `TableRow`.

Шаг 8. Добавление компонентов в `tableRow1`

Добавьте три компонента `TextView` в объект `tableRow1`, выполнив следующие действия:

1. Перетащите компонент `TextView` (`tenTextView`) на объект `tableRow1`, находящийся в окне `Outline`.
2. Повторите этот процесс, добавив компоненты `fifteenTextView` и `twentyTextView`.
3. Установите значения свойств `Id` и `Text` для каждого компонента.

Шаг 9. Добавление компонентов в `tableRow2`

Добавьте компонент `TextView` и три компонента `EditText` в `tableRow2`, выполнив следующие действия:

1. В окне `Outline` перетащите компонент `TextView` (`tipTextView`) на `tableRow2`.
2. В окне `Outline` перетащите три компонента `EditText` на `tableRow2` (`tip10EditText`, `tip15EditText` и `tip20EditText`).
3. Установите значения свойств `Id` и `Text` для каждого компонента.

Шаг 10. Добавление компонентов в `tableRow3`

Добавьте компонент `TextView` и три компонента `EditText` в `tableRow3`, выполнив следующие действия:

1. В окне `Outline` перетащите компонент `TextView` (`totalTextView`) на `tableRow3`.
2. В окне `Outline` перетащите три компонента `EditText` на `tableRow3` (`total10EditText`, `total15EditText` и `total20EditText`).
3. Установите значения свойств `Id` и `Text` для каждого компонента.

Шаг 11. Добавление компонентов в tableRow4

Добавьте компоненты `TextView`, `SeekBar` и другой компонент `TextView` в `tableRow4`, выполнив следующие действия:

1. В окне `Outline` перетащите компонент `TextView` (`customTextView`) на `tableRow4`.
2. В окне `Outline` перетащите компонент `SeekBar` (`customSeekBar`) на `tableRow4`.
3. В окне `Outline` перетащите компонент `TextView` (`customTipTextView`) на `tableRow4`.
4. Установите значения свойств `Id` и `Text` для компонентов `TextView`.

Шаг 12. Добавление компонентов в tableRow5

Добавьте компонент `TextView`, `EditText`, другой компонент `TextView` и еще один компонент `EditText` в `tableRow5`, выполнив следующие действия:

1. В окне `Outline` перетащите компонент `TextView` (`tipCustomTextView`) на `tableRow5`.
2. В окне `Outline` перетащите компонент `EditText` (`tipCustomEditText`) на `tableRow5`.
3. В окне `Outline` перетащите компонент `TextView` (`totalCustomTextView`) на `tableRow5`.
4. В окне `Outline` перетащите компонент `EditText` (`totalCustomEditText`) на `tableRow5`.
5. Установите значения свойств `Id` и `Text` для каждого компонента.

4.4.3. Просмотр созданного макета

Начиная с этого момента графический интерфейс пользователя будет выглядеть так, как показано на рис. 4.4.

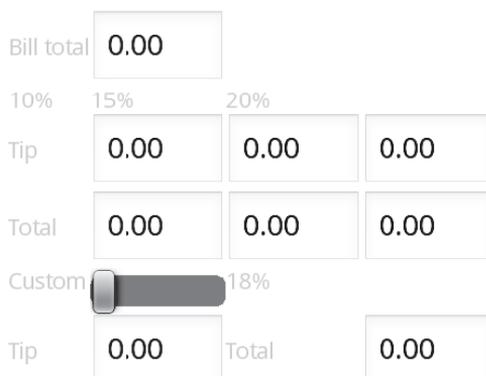


Рис. 4.4. Графический интерфейс пользователя `Tip Calculator` перед настройкой свойств, отличающихся от свойств `Id` и `Text` для каждого компонента

При сравнении с рис. 4.2 заметны следующие различия.

- Компоненты `billEditText` и `customSeekBar` еще не охватывают несколько столбцов.
- Текст для всех компонентов `TextView` выделен светло-серым цветом, что затрудняет его чтение.