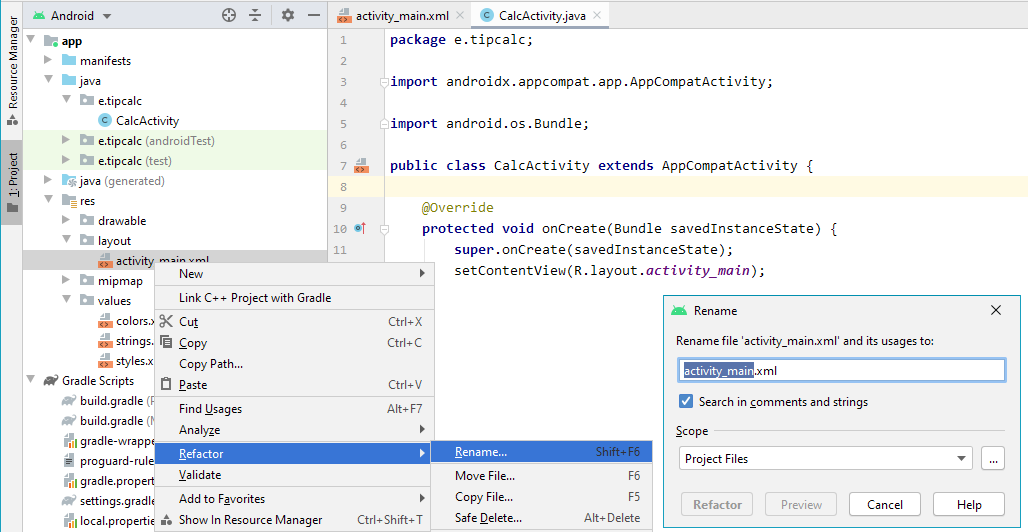
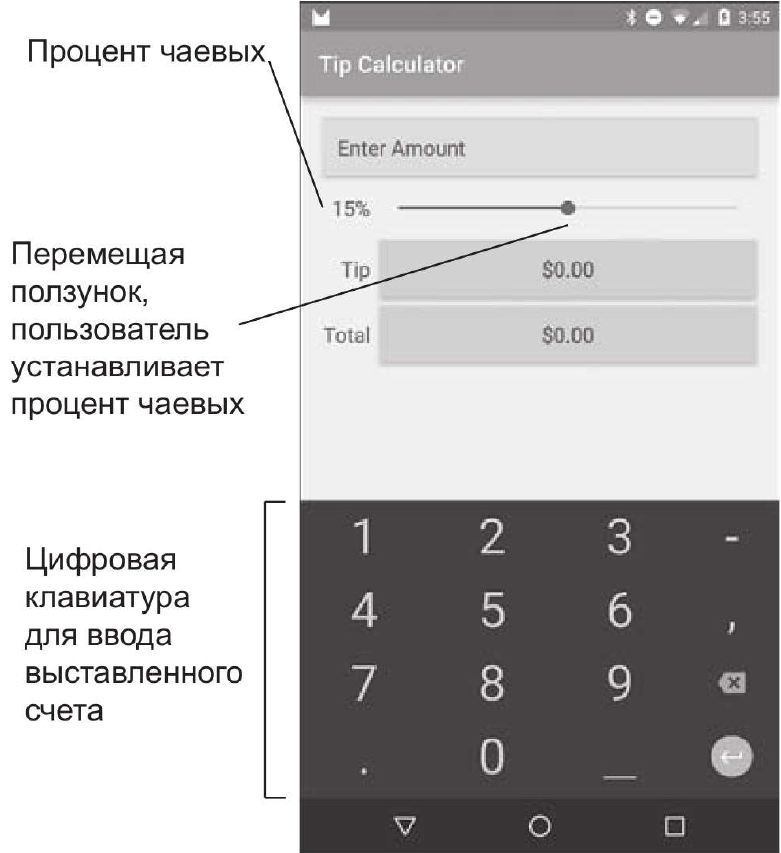
**Создание приложения TipCalculator**

1. Создать приложение (**EmptyActivity**) (*Error:* DarkActionBar на **NoActionBar**)
2. Выполните **рефакторинг** и замените название файла *MainActivity.java* на **CalcActivity.java** и наименование файла макета *activity\_main.xml* соответственно на **calc\_activity.xml.**Для этого, выделите *MainActivity* и, правой кнопкой мыши вызовите контекстное меню, в котором выберите **Refactor**, а далее **Rename** (либо нажмите клавиши *Shift+F6*). Введите **CalcActivity**.  
   Также можно щёлкнуть правой кнопкой мыши на имени файла в окне проекта и, вызвав контекстное меню, выбрать**Refactor**, а далее**Rename**.
3. Аналогично замените *activity\_main.xml* на **calc\_activity.xml**:  
   
4. Перейдите на файл макета **calc\_activity.xml** в режиме **Desing** и добавьте с палитры **Palette**на дерево компонент (**ComponentTree**) макета следующие поля:

Сумма счёта

|  |  |
| --- | --- |
| **Свойство** | **Значение** |
| **PlainText(EditText)** |  |
| id | et\_amount |
| text | Enter amount |
| hint | Input text |
| inputType | numberDecimal |
| selectAllOnFocus | true |

Процентчаевых

|  |  |
| --- | --- |
| **Свойство** | **Значение** |
| **TextView** |  |
| id | tv\_percent |
| text | 15% |

Ползунок

|  |  |
| --- | --- |
| **Свойство** | **Значение** |
| **SeekBar** |  |
| id | sb\_percent |
| layout\_width | 0dp |

Метка для чаевых

|  |  |
| --- | --- |
| **Свойство** | **Значение** |
| **TextView** |  |
| id | tv\_label\_tip |
| text | Tip |

Сумма чаевых

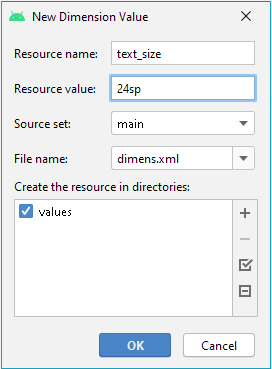
|  |  |
| --- | --- |
| **Свойство** | **Значение** |
| **TextView** |  |
| id | tv\_tip |

Метка для итоговой суммы

|  |  |
| --- | --- |
| **Свойство** | **Значение** |
| **TextView** |  |
| id | tv\_label\_total |
| text | Total |

Итоговая сумма

|  |  |
| --- | --- |
| **Свойство** | **Значение** |
| **TextView** |  |
| id | tv\_total |

1. ***Предупреждение***: Компоненты в макете **ConstraintLayout** нельзя перемещать мышкой. Размещать их надо, только цепляя **«пружинки»**(**constraint**) друг к другу за кругляшки или к краям экрана.
2. Задайте размер текста для всех полей: **android:textSize = "24sp"**. Для этого выделите один из компонентов, в панели **Attributes** найдите свойство **textSize** , в конце строки атрибута нажмите мышкой значок 0 (овал). Откроется диалоговое окно с кнопкой **AddDimensionValue**, нажав на которую откроется следующее окно для задания имени и значения константы. Задайте имя**: text\_size**, а значение **24sp**. Нажмите кнопку **Ok,** и будет создан файл **dimens.xml**, в котором будут хранится константы для размеров различных компонент макета.  
   Для остальных компонент в атрибутах **textSize** укажите имя только что созданной константы: **@dimen/text\_size**.
3. Задайте свойства компонент в соответствии с приведённой выше таблицей в окне атрибутов справа. Все текстовые константы должны быть описаны в файле **strings.xml**. Текстовые константы определяются аналогично константам размеров.
4. Запустите приложения и проверьте правильность макета.
5. Теперь вернёмся к коду приложения.
6. Создаём класс Calc. Для этого в окне проекта щёлкаем правой кнопкой мыши на разделе java/”имя проекта”, в открывшемся меню выбираем **New**, далее **JavaClass**. В окне **ChooseDestinationDirectory** необходимо проверить, что выбрана папка **…/main/java**. В следующем окне **NewJavaClass** в строке Name введите имя класса **Calc**.
7. Описываем переменные класса:

**private double billAmount**; *// Сумма счёта*  
**private double percent**; *// Процент чаевых*  
Методы класса: (Alt+Insert)

**public double** calculateTip(**double** billAmount, **double** percent) { **return** billAmount\*percent; }  
**public double** calculateTotal(**double** billAmount, **double** percent) { **return** billAmount\*(1+percent);}

// **Getter and setter**:

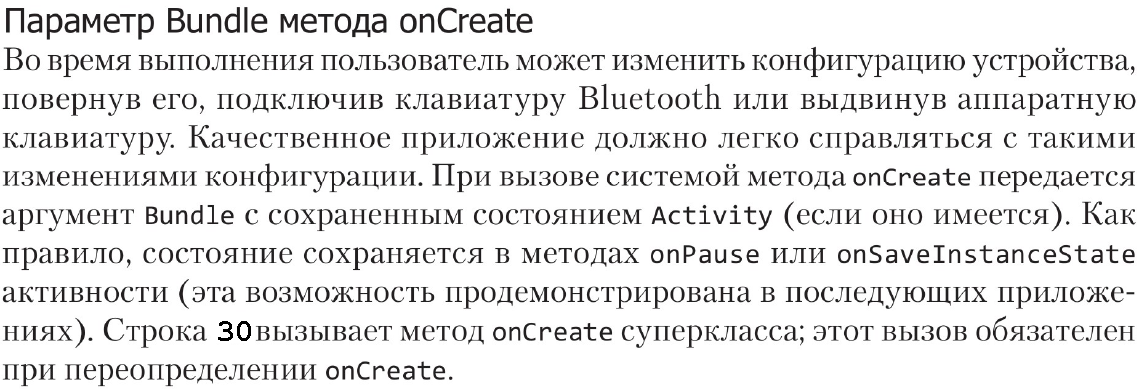
**public double** getBillAmount() { **return billAmount**; }  
**public void** setBillAmount(**double** billAmount) { **this**.**billAmount** = billAmount; }  
  
**public double** getPercent() { **return percent**; }  
**public void** setPercent(**double** percent) { **this**.**percent** = percent; }

1. Возвращаемся в класс **CalcActivity** и описываем переменные:

**Private double amount**= 0.0; *// Сумма счёта***private double percent**= 0.15; *// Процент чаевых по умолчанию.***private** EditText **et\_amount**; *// Поле для ввода суммы счёта***private** SeekBar **sb\_percent**; *// Ползунок для процентов***private** TextView **tv\_percent**; *// Поле для значения процента***private** TextView **tv\_tip**; *// Поле для суммы чаевых***private** TextView **tv\_total**; *// Поле для итоговой суммы*

1. Создайте экземпляр класса Calc:  
   *// Создание объекта калькулятор чаевых:***private** Calc **tipCalc** = **new** Calc();
2. Метод **onCreate** вызывается при создании Activity.

Аргумент **Bundle** передаёт сохранённое состояние:



1. В методе **onCreate** связываем idView в макете с соответствующими переменными в коде приложения:

**et\_amount** = findViewById(R.id.***et\_amount***);  
**sb\_percent** = findViewById(R.id.***sb\_percent***);  
**tv\_percent** = findViewById(R.id.***tv\_percent***);  
**tv\_tip** = findViewById(R.id.***tv\_tip***);  
**tv\_total** = findViewById(R.id.***tv\_total***);  
// Для текстовых полей задаём первоначальные значения  
**tv\_tip**.setText(**"0.0"**);  
**tv\_total**.setText(**"0.0"**);

1. Слушатели (**Listener**) реагирует на события интерфейса. В этом приложении надо обработать ввод текста и перемещения ползунка. Для этого создаём **Слушателей** (Javapattern) для редактируемого поля (EditText) и ползунка (SeekBar):

**et\_amount**.addTextChangedListener(**amountTextWatcher**);  
**sb\_percent**.setOnSeekBarChangeListener(**sbListener**);

1. Завешаем метод **onCreate**

}

1. Переопределяем методы слушателя TextWatcher:

*// Интерфейс слушателя изменений текста в EditText***private final** TextWatcher **amountTextWatcher** = **new** TextWatcher() {  
*//* ***Вызывается при изменении пользователем величины счета***@Override  
**public void** onTextChanged(CharSequence s, **int** start, **int** before, **int** count) {  
**amount** = Double.*parseDouble*(s.toString());  
*// Обновление полей с чаевыми и общей суммой***tv\_tip**.setText(Double.*toString*(**tipCalc**.calculateTip(**amount**,**percent**)));  
**tv\_total**.setText(Double.*toString*(**tipCalc**.calculateTotal(**amount**, **percent**)));  
 }  
@Override  
**public void** afterTextChanged(Editable s) { }  
@Override  
**public void** beforeTextChanged(  
 CharSequence s, **int** start, **int** count, **int** after) { }

};

1. Переопределяем методы слушателя OnSeekBarChangeListener:

*// Интерфейс слушателя изменений состояния SeekBar***private final** OnSeekBarChangeListener **sbListener**=

**new**OnSeekBarChangeListener() {  
*// Обновление процента чаевых и итоговой суммы*@Override  
**public void** onProgressChanged(SeekBarseekBar, **int** progress, **boolean** fromUser) {  
**percent**= progress / 100.0; *// Назначение процента чаевых  
 // Вычисление чаевых и общей суммы. Вывод их на экран.***tv\_percent**.setText(Double.*toString*(**percent**));  
**tv\_tip**.setText(Double.*toString*(**tipCalc**.calculateTip(**amount**, **percent**)));  
**tv\_total**.setText(Double.*toString*(**tipCalc**.calculateTotal(**amount**, **percent**)));  
 }  
@Override  
**public void** onStartTrackingTouch(SeekBar seekBar) { }  
@Override  
**public void** onStopTrackingTouch(SeekBar seekBar) { }  
};

1. *Замечание*: Обратите внимание, что слушатели реализуются как анонимные классы и могут быть заменены Лямда-функцией (->). Раскрыть лямда-функцию можно, нажав на + в начале строки.
2. Запустите приложение и **обратите внимание на вывод данных** в текстовых полях.
3. Данные нужно отформатировать. Добавьте строки в начало **CalcActivity**:

*// Форматирование денежных сумм и процентов***private static final** NumberFormat ***currencyFormat***= NumberFormat.*getCurrencyInstance*();  
**private static final** NumberFormat ***percentFormat*** = NumberFormat.*getPercentInstance*();

1. Измените все строки, где задаются значения текстовых полей **setText** следующим образом:

**tv\_percent**.setText(***percentFormat***.format(**percent**));  
**tv\_tip**.setText(***currencyFormat***.format(**tipCalc**.calculateTip(**amount**,**percent**)));  
**tv\_total**.setText(***currencyFormat***.format(**tipCalc**.calculateTotal(**amount**, **percent**)));

1. Ресурсы (сменить иконку приложения ic\_launcher).
2. Выполнить локализацию для русского языка.
3. Переключите настройки языка и проверьте, что локализация работает.