

Сергеев Павел Андреевич

студент

ФГБОУ ВО «Государственный университет морского
и речного флота им. адмирала С.О. Макарова»

г. Санкт-Петербург

РАЗРАБОТКА МОБИЛЬНОГО ПРИЛОЖЕНИЯ-КАЛЬКУЛЯТОРА

***Аннотация:** данная статья посвящена работе над созданием визуального интерфейса и программного кода мобильного приложения «Calculator» для Android. Разрабатываемый калькулятор похож на обычный системный калькулятор, он также может работать с очень большими и очень маленькими значениями, которые поддерживаются классом объектов из языка программирования Java – *BigDecimal*.*

***Ключевые слова:** Android, Java, калькулятор, программирование для мобильных систем.*

Краткий обзор

Целью работы было создание приложения, которое выполняло бы функцию простого калькулятора, имело бы визуально простое устройство, и могло бы работать с широким диапазоном значений.

Экран приложения состоит из двух макетов *LinearLayout*, из которых в одном расположено поле *TextView*, где отображаются расчеты, а в другом несколько контейнеров *RadioButton*, в которые сгруппированы элементы *Button* (кнопки). Приложение представляет собой простой калькулятор с двумя режимами работы: вертикальным (Рисунок 1) и горизонтальным (Рисунок 2). Разработанный визуальный интерфейс представлен в конце статьи.

В вертикальном режиме доступны: кнопки ввода цифр и запятой, кнопка «%» (процент), кнопка «+/-» (плюс-минус), кнопка «C» (стирание последнего введенного числа), кнопки основных арифметических знаков (сумма, разность, произведение и деление), кнопка равенства и кнопка «AC» (стирание всех введенных данных). В горизонтальном режиме, помимо вышеперечисленных

кнопок, доступны: кнопка «1/x» (деление единицы на введённое значение), кнопка "!" (факториал введённого числа), кнопки констант «пи» и «е», кнопки возведения введённого значения во вторую и третью степень, кнопки вычисления натурального и десятичного логарифма из введённого значения, кнопки вычисления квадратного и кубического корней из введённого значения, кнопки возведения числа 10 в степень введённого значения и кнопки, дающей на выводе случайное число от 0 до 1.

Механика работы проста и не сильно отличается от используемых в других калькуляторах.

Алгоритм работы приложения

Алгоритм работы представляет собой взаимодействие двух значений, которые в программном коде объявлены как `firstNumberBD` и `ResultBD` (добавление BD означает принадлежность объектов к классу `BigDecimal`). Для суммирования двух значений, к примеру, необходимо: ввести первое значение, нажать на соответствующий арифметический знак, затем ввести второе значение и нажать на кнопку равенства (либо ещё раз на кнопку с выбранным знаком, это сделано для облегчения работы, в процессе которой используется один арифметический знак).

Нажатие на кнопку «%» (процент) умножает введённое значение на 0,01. Кнопка «+/-» меняет знак перед введённым значением на противоположный. По такому же механизму работают кнопки возведения в степень 2 или 3, извлечения корня степени 2 или 3, взятие натурального и десятичного логарифма, и факториал.

Каждой кнопке на экране приложения соответствует некоторый метод, который отвечает за восприятие нажатия на неё. В языке разметки XML это прописывается следующим образом – `android:onClick= «название используемого метода»`, данная строка помещается внутрь дочернего элемента, то есть `Button`. Сам метод должен находиться внутри используемого класса Java.

Ввод значений

Кнопки с цифрами от 0 до 9 имеют общий метод, который управляет набором значения и его записью в соответствующую переменную (объект класса). Данный метод был назван `onClickNumber`.

Процесс обработки нажатия на любую из цифр одинаков для всех кнопок. Сначала метод проверяет то, на какую кнопку было произведено нажатие, за это отвечает конструкция `switch-case` (выбор идёт по `id` элемента разметки XML). Затем идёт проверка на то, идёт ли ввод цифр после запятой или перед ней (для этого были введены булевая переменная `CommaMode` и переменная типа `int NumberPostComma`). В конце нужная цифра добавляется к уже имеющемуся (или нет) числу (цифре).

Процесс добавления цифры к объекту достаточно интересный, имеющиеся объекты принадлежат классу `BigDecimal`, который позволяет работать со своим объектом как со строкой, и со строкой как со своим объектом. Об этой и других возможностях речь пойдёт ниже.

Класс `BigDecimal`

В задаче с калькулятором, где самое важное – точность, переменные примитивных типов `float` и `double` не смогут обеспечить наилучшую возможную точность расчётов. Взять, например, результат разности двух переменных типа `double` со значениями 2 и $1.1 - 0.8999...$. Подобных проблем и ограничений в точности можно избежать, используя объекты класса `BigDecimal`.

Класс `BigDecimal` позволяет создавать объекты, которые являются числами с плавающей точкой, и которые могут быть заданы с произвольной точностью. Особенность данного класса также состоит в том, что его объекты являются неизменяемыми. Поэтому на каждом шаге работы со значениями в данном калькуляторе эти объекты создаются заново, изменяясь по заданному сценарию.

Процесс создания объекта данного класса для Java выглядит обычно – `BigDecimal a = new BigDecimal();`. Данный класс позволяет создавать объекты, присваивая им строковые значения – `BigDecimal a = new BigDecimal("100");`.

Данный класс обладает большим количеством математических функций. Основные математические функции, которые предоставлены в качестве методов используемым классом, – `add` (сумма), `subtract` (разность), `multiply` (произведение) и `divide` (деление) дают возможность устанавливать в теле своих методов необходимое значение округления в виде константы или переменной типа `int`, а также тип округления (в данном калькуляторе использовался тип `BigDecimal.ROUND_HALF_DOWN`, который используется и в обычной школьной арифметике). Также доступны некоторые константы, самая используемая в данном калькуляторе – `BigDecimal.ZERO` (т.е. ноль).

В некоторых случаях к расчётам были подключены компоненты библиотеки `java.lang.Math` для расчёта математических функций (например, вычисление натурального логарифма) со значениями типа `double`. Данное отступление от использования объектов только класса `BigDecimal` обусловлено сложностью найденных в процессе разработки методов для данного класса, процессор тестируемого устройства нагружался в некоторых случаях до 100%.

Поворот экрана и сохранение введённых значений

С помощью переопределённого метода `onSaveInstanceState`, который принимает на вход переменную-контейнер класса `Bundle` – `outState`, происходит сохранение необходимых для калькулятора значений (`firstNumberBD`, `ResultBD` и др.). Значения кладутся в контейнер соответствующими для типа значений методами. Для каждого введённого в контейнер значения необходимо уникальное строковое имя, и обращаясь к которому, в необходимых случаях происходит изменение данных значений или их изъятие для дальнейшей с ними работы.

При помощи данного контейнера осуществляется сохранение и возможность отображения введённых значений, к примеру, при повороте экрана. Непосредственно контроль отображения и присвоения значений при повороте берёт на себя метод `transferMethod`.

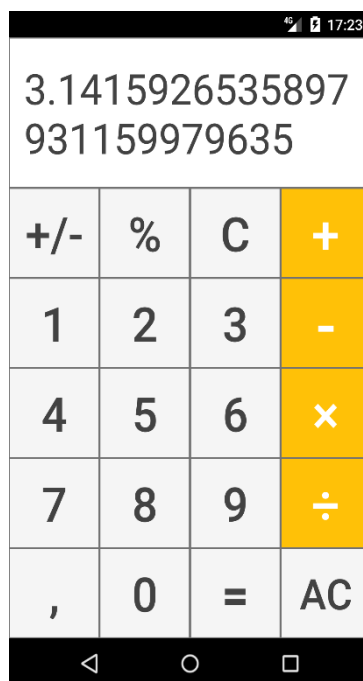
Визуальный интерфейс приложения

Рис. 1. Вертикальная ориентация приложения

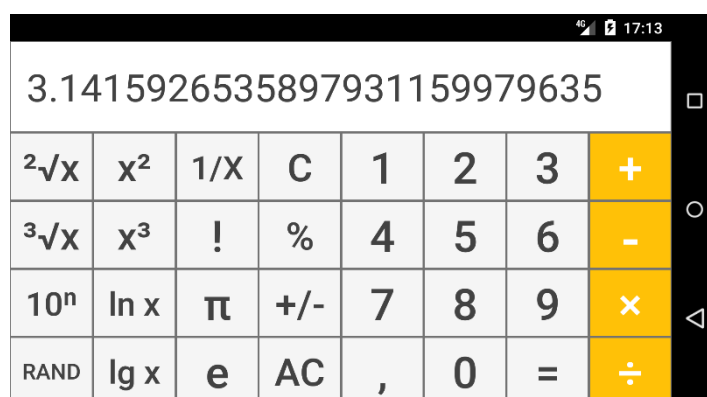


Рис. 2. Горизонтальная ориентация приложения

Заключение

Разработанное приложение было протестировано и загружено в GooglePlay (<https://play.google.com/store/apps/details?id=app.pavel.calculator>) и на GitHub (<https://github.com/PavelSergeev1/Calculator>), где представлен весь описанный программный код приложения, в том числе и сведения о строковых, цветовых, стилевых и других ресурсах, которые также были подобраны и описаны в процессе разработки.

На данный момент готовится второе по счёту обновление, в котором будет расширен функционал нескольких кнопок и доработан их режим работы.

Список литературы

1. [Электронный ресурс]. – Режим доступа: <https://docs.oracle.com/javase/7/docs/api/java/lang/Math.html>
2. [Электронный ресурс]. – Режим доступа: <https://docs.oracle.com/javase/7/docs/api/java/math/BigDecimal.html>
3. [Электронный ресурс]. – Режим доступа: <https://developer.android.com/reference/android/widget/LinearLayout>
4. [Электронный ресурс]. – Режим доступа: <https://developer.android.com/reference/android/widget/TextView>
5. Программирование под Android / З. Медникс, Л. Дорнин, Б. Мик. – 2-е изд.