

## Введение

К нашей компании обратился основатель Microsoft Билл Гейтс с проблемой калькулятора в OS Windows 11: “Our calculator is extremely inconvenient to use, no one understands how to multiply or divide on it; users are moving to Linux or Mac OS. Help us urgently! The budget is unlimited!”.

Проведя совещание, длительностью шесть секунд, мы согласились ему помочь, а учитывая все обстоятельства – убедили из всех арифметических операций оставить только сложение.

Было составлено и утверждено следующее техническое задание.

### Техническое задание «Калькулятор»

#### Основные требования

1. Использовать Clean Architecture + MVP (Model-View-Presenter);
2. Поддерживаемые арифметические операции - сложение;
3. Состояние приложения сохраняется между сеансами.

#### Использовать Clean Architecture + MVP?

Реализовать слои Presentation/Data/Domain, каждый из которых находится в своей assembly. Presentation слой реализовать с использованием шаблона Model-View-Presenter.

#### Поддерживаемые арифметические операции - сложение?

В случае, если пользователь вводит что-то кроме чисел и знака «+», в результат выводить сообщение «Error». Примеры **правильных** выражений:  $54+21$ ,  $45+00$ . Примеры **неправильных** выражений:  $45+-88$ ,  $98.12+48.1$ . Сделать нужно так, как получается проще всего. Главное получить результат сложения введенных пользователем данных или вывести сообщение «Error», если результат получить невозможно.

#### Состояние приложения сохраняется между сеансами?

Когда пользователь закрывает приложение, нужно сохранять его состояние, в данном случае - введенное пользователем выражение. После открытия приложения, сохраненное состояние нужно восстановить. Пример: пользователь ввел  $34+47$ , закрыл приложение, открывает его, на экране введено  $34+47$ .

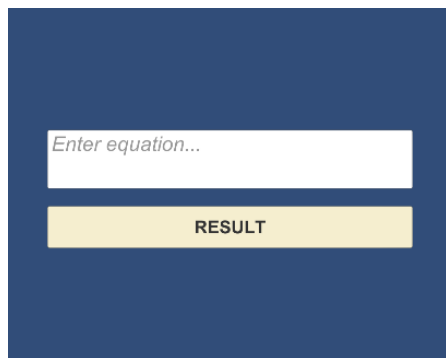


Рисунок 1 – пример UI приложения

#### Сценарий использования

Пользователю доступно поле ввода и кнопка «Result», по нажатию на которую в поле ввода выводится результат выражения. Если пользователь вводит что-то кроме чисел и знака «+», то в результат выводится сообщение «Error» (Основные требования п.2). При открытии приложения, пользователь продолжает работу с того же места, где он и остановился (Основные требования п.3). Пример UI приложения можно увидеть на рисунке 1.

## **Результат**

Результат выполнения технического задания предоставить в виде исходного кода (zip или unitypackage архив, ссылка на проект github или bitbucket, в зависимости от предпочтений исполнителя).

Данное техническое задание защищено авторским правом и честным словом. Всем приступающим к его выполнению желаем успехов и приятного время провождения.