

Министерство образования и молодежной политики Свердловской области

ГАПОУ СО «Екатеринбургский колледж транспортного строительства»

Отчёт по учебной практике

УП 01.02

Выполнил: Епимахов Денис Леонидович

Группа: ПР-21

Преподаватель: Мирошниченко Г.В.

2025

Содержание

[1. **Задание №1 Мобильное приложение «Кредитный калькулятор»** 3](#_Toc118960339)

[1.1 Описание задачи 3](#_Toc118960340)

[1.2 Структура проекта 4](#_Toc118960341)

[1.3 Описание разработанных функций 5](#_Toc118960342)

[1.4 Используемые библиотеки 9](#_Toc118960344)

[1.5 Тестовые случаи 9](#_Toc118960345)

[1.6 Используемые инструменты 20](#_Toc118960346)

[1.7 Описание пользовательского интерфейса 20](#_Toc118960347)

[1.8 Приложение 21](#_Toc118960348)

# **1. Задание №3 мобильное приложение «Кредитный калькулятор»**

## **Описание задачи**

Используя различные методы создать мобильное приложение для расчета кредита используя:

* Навигацию TabbedPage;
* Поля Entry;
* Slider;
* Поля Label;
* Picker;
* DatePicker;
* Стилизацию через css файл;
* Методы для проверок входных данных;
* Разработать модульные тесты к реализованным методам классов;

## **Структура проекта**

Решение состоит из 3 проектов (Рис. 1): zd4\_EpimakhovDenis, TestAPP и zd4\_EpimakhovDenis.Android. Основной проект zd4\_EpimakhovDenis содержит: экраны с классом – Page1, Page 2 и WelcomePage; класс с логикой – CreditCalculator; файл со стилям – styles.css. Проект TestAPP представляет из себя проект тестов, содержащую 20 модульных тестов в классе CreditCalculator.

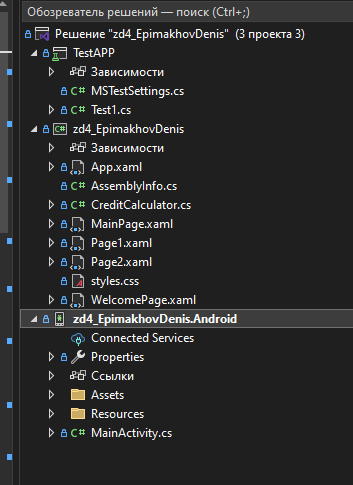


Рис. 1. Обозреватель решений

## **1.3 Описание разработанных функций**

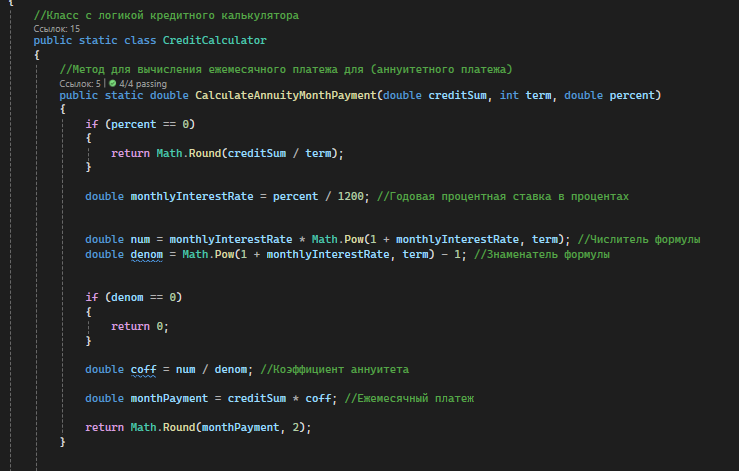
 Класс (CreditCalculator.cs) с логикой состоит из 5 методов

Рис. 2. Метод для вычисления ежемесячного платежа для (аннуитетного платежа).

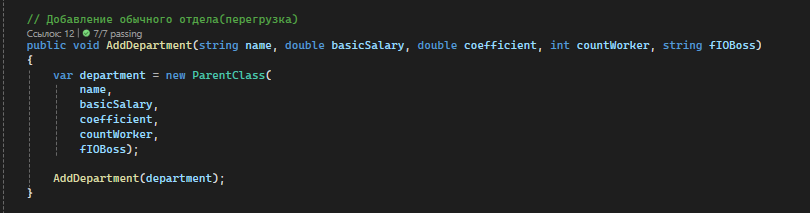


Рис. 3. Метод для вычисления общей суммы для (аннуитетного платежа).

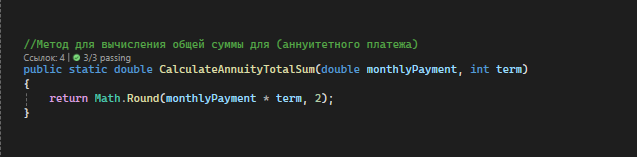


Рис. 4. Метод для вычисления общей суммы для (аннуитетного платежа).

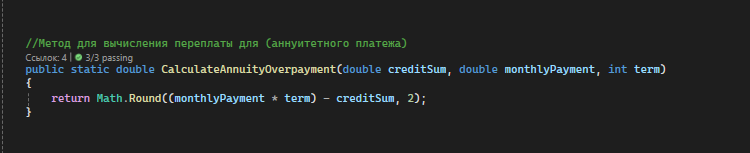


Рис. 5. Метод для вычисления переплаты для (аннуитетного платежа).

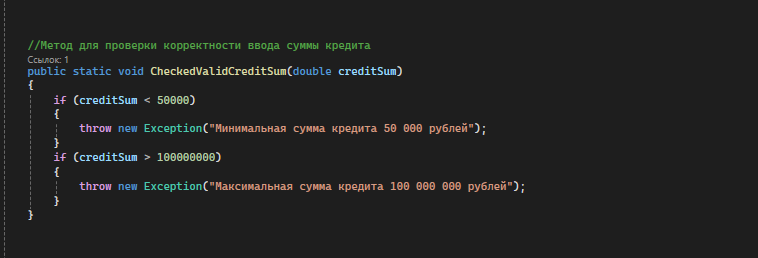


Рис. 6. Метод для проверки корректности ввода суммы кредита.

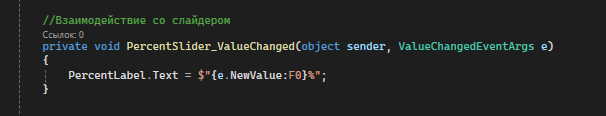
Класс (Page1.cs) с логикой состоит из 2 методов

Рис. 7. Взаимодействие со слайдером.

Рис. 8. Кнопка рассчитать.

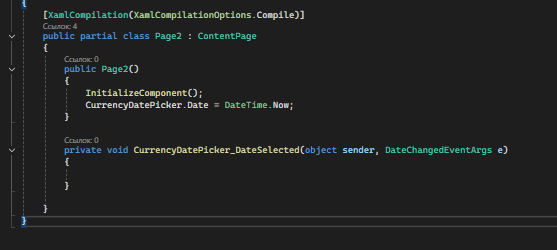
Класс (Page2.cs)

Рис. 9. DatePicker.

## **1.4 Используемые библиотеки**

* using System;
* using System.Collections.Generic;
* using System.Linq;
* using System.Text;
* using System.Threading.Tasks;
* using Xamarin.Forms;
* using Xamarin.Forms.Xaml;System.Windows.Forms.VisualStyles.VisualStyleElement.StartPanel;

## **1.5 Тестовые случаи**

Проект TestAPP состоит из 20 тестов. Тесты нужны для проверки правильной работоспособности методов.

Название теста: TestMethod1;

Назначение теста: Проверка стандартного расчета ежемесячного аннуитетного платежа (Метод - CalculateAnnuityMonthPayment(credit, term, percent);

Входные данные: double credit = 10000; int term = 12; double percent 20;

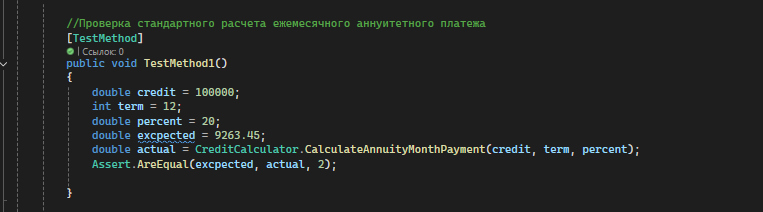
Ожидаемый результат: double excpected = 8357.66;

Рис. 10.

Название теста: TestMethod2;

Назначение теста: Минимальная сумма кредита и минимальный срок (Метод - CalculateAnnuityMonthPayment(credit, term, percent);

Входные данные: double credit = 50000; int term = 6; double percent 1;

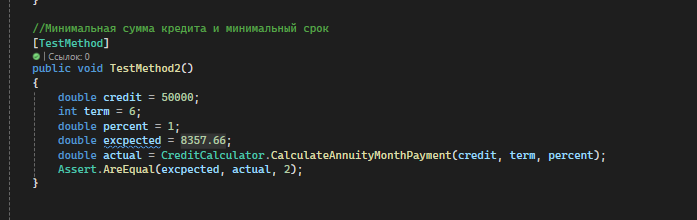
Ожидаемый результат: double excpected = 8357.66;

Рис. 11.

Название теста: TestMethod3;

Назначение теста: Максимальная сумма кредита и максимальный срок (Метод - CalculateAnnuityMonthPayment(credit, term, percent);

Входные данные: double credit = 100000000; int term = 360; double percent = 25;

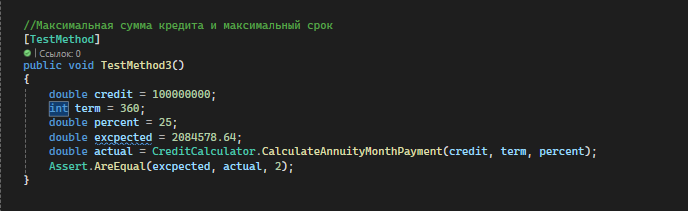
Ожидаемый результат: double excpected = 2084578.64;

Рис. 12.

Название теста: TestMethod4;

Назначение теста: Процентная ставка равна нулю (Метод - CalculateAnnuityMonthPayment(credit, term, percent);

Входные данные: double credit = 60000; int term = 24; double percent = 0;

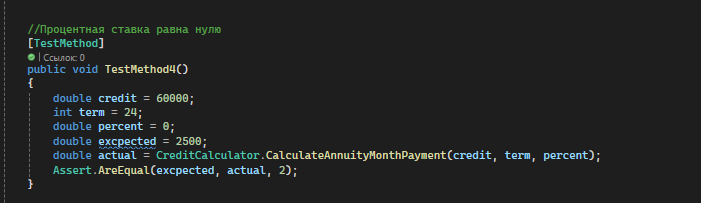
Ожидаемый результат: double excpected = 2500;

Рис. 13.

Название теста: TestMethod5;

Назначение Стандартный расчет общей суммы выплат (Метод - CalculateAnnuityTotalSum(monthlyPayment, term);

Входные данные: double credit = 2500; int term = 24;

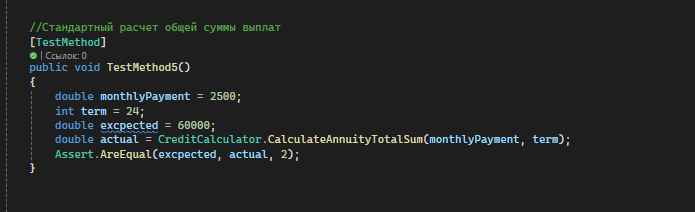
Ожидаемый результат: double excpected = 60000;

Рис. 14.

Название теста: TestMethod6;

Назначение теста: Минимальные значения для расчета общей суммы (Метод - CalculateAnnuityTotalSum(monthlyPayment, term);

Входные данные: double credit = 60000; int term = 1;

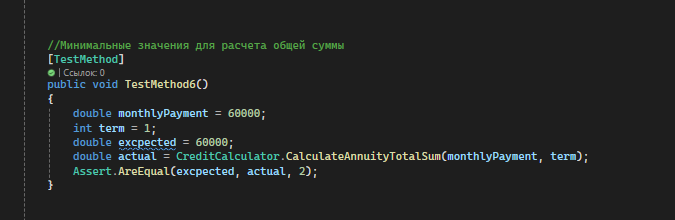
Ожидаемый результат: double excpected = 60000;

Рис. 15.

Название теста: TestMethod7;

Назначение теста: Максимальные значения для расчета общей суммы (Метод - CalculateAnnuityTotalSum(monthlyPayment, term);

Входные данные: double credit = 15000000; int term = 360;

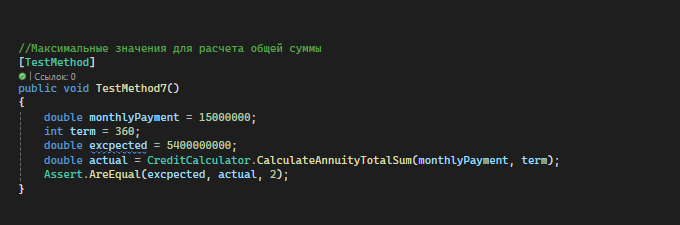
 Ожидаемый результат: double excpected = 5400000000;

Рис. 16.

Название теста: TestMethod8;

Назначение теста: Стандартный расчет переплаты по кредиту (Метод - CalculateAnnuityOverpayment(creditSum, monthlyPayment, term);

Входные данные: double creditSum = 15000000; double monthlyPayment = 2024936.55; int term = 12;

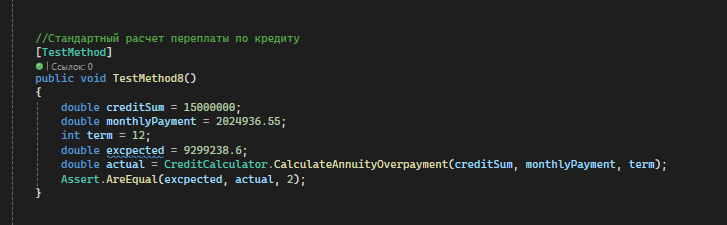
 Ожидаемый результат: double excpected = 9299238.6;

Рис. 17.

Название теста: TestMethod9;

Назначение теста: Переплата при нулевой процентной ставке (Метод - CalculateAnnuityOverpayment(creditSum, monthlyPayment, term);

Входные данные: double creditSum = 200000; double monthlyPayment = 16667; int term = 12;

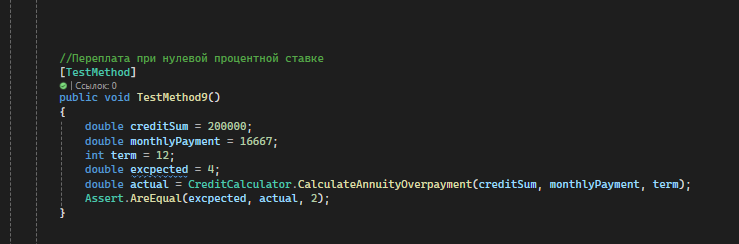
 Ожидаемый результат: double excpected = 4;

Рис. 18.

Название теста: TestMethod10;

Назначение теста: Максимальные значения для расчета переплаты (Метод - CalculateAnnuityOverpayment(creditSum, monthlyPayment, term);

Входные данные: double creditSum = 100000000; double monthlyPayment = 1681708.33; int term = 360;

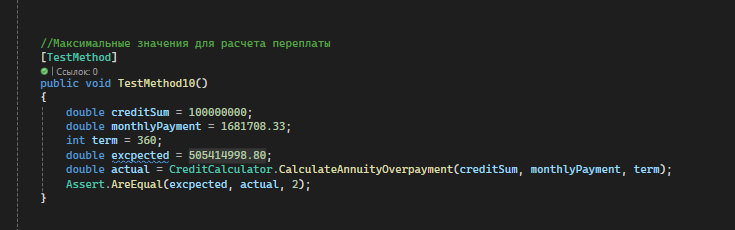
 Ожидаемый результат: double excpected = 505414998.80;

Рис. 19.

Название теста: TestMethod11;

Назначение теста: Проверка минимальной суммы кредита (меньше 50 000) (Метод - CheckedValidCreditSum(credit));

Входные данные: double credit = 25000;

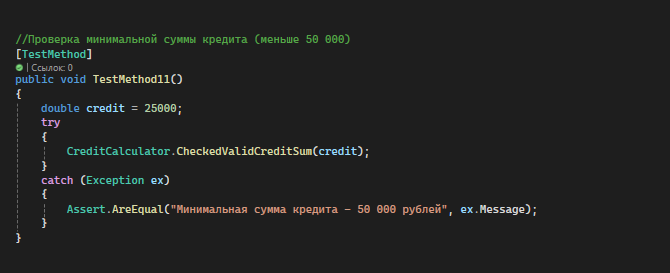
 Ожидаемый результат: "Минимальная сумма кредита - 50 000 рублей ";

Рис. 20.

Название теста: TestMethod12;

Назначение теста: Проверка максимальной суммы кредита (больше 100 000 000) (Метод - CheckedValidCreditSum(credit));

Входные данные: Входные данные: double credit = 10000000000000;

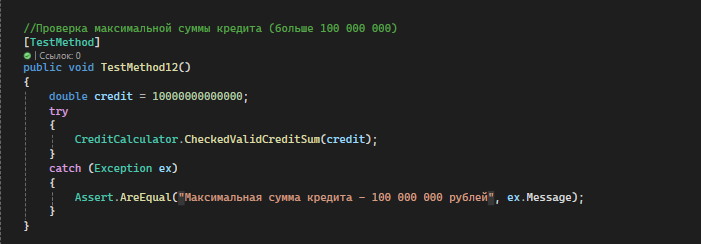
 Ожидаемый результат: "Максимальная сумма кредита - 100 000 000 рублей";

Рис. 21.

Название теста: TestMethod13;

Назначение теста: Средняя корректная сумма кредита (5 000 000.50) (Метод - CheckedValidCreditSum(credit));

Входные данные: Входные данные: double credit = 5000000.50;

 Ожидаемый результат: выполнение метода (CheckedValidCreditSum(credit));

Рис. 22.

Название теста: TestMethod14;

Назначение теста: Точное значение максимальной суммы кредита (100 000 000) (Метод - CheckedValidCreditSum(credit));

Входные данные: double credit = 100000000;

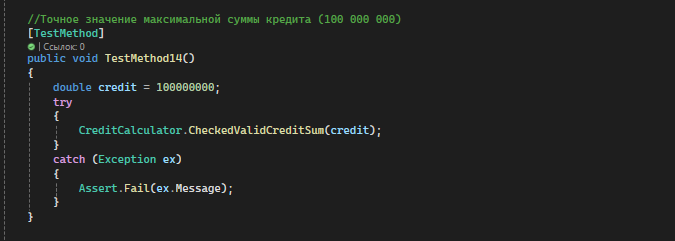
 Ожидаемый результат: выполнение метода (CheckedValidCreditSum(credit));

Рис. 23.

Название теста: TestMethod15;

Назначение теста: Точное значение минимальной суммы кредита (50 000) (Метод - CheckedValidCreditSum(credit));

Входные данные: double credit = 50000;

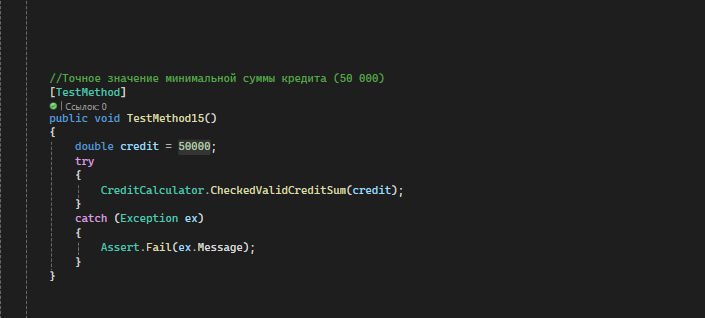
 Ожидаемый результат: выполнение метода (CheckedValidCreditSum(credit));

Рис. 24.

Название теста: TestMethod16;

Назначение теста: Некорректный срок — меньше минимального (меньше 6 месяцев) (Метод - CheckedValidTerm(term));

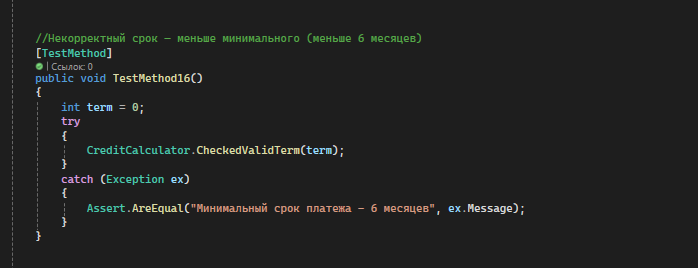
 Ожидаемый результат: "Минимальный срок платежа - 6 месяцев";

Рис. 25.

Название теста: TestMethod17;

Назначение теста: Некорректный срок — больше максимального (больше 360 месяцев) (Метод - CheckedValidTerm(term));

Входные данные: int term = 365;

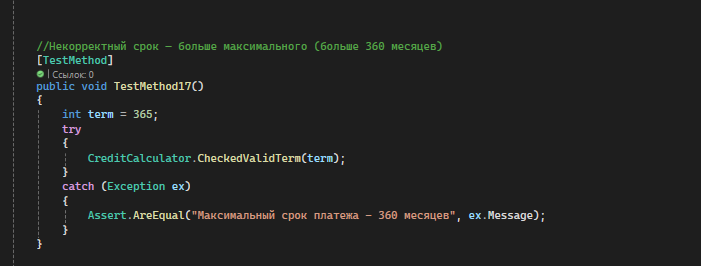
 Ожидаемый результат: "Максимальный срок платежа - 360 месяцев";

Рис. 26.

Название теста: TestMethod18;

Назначение теста: Корректный срок — среднее значение (24 месяца) (Метод - CheckedValidTerm(term));

Входные данные: int term = 24;

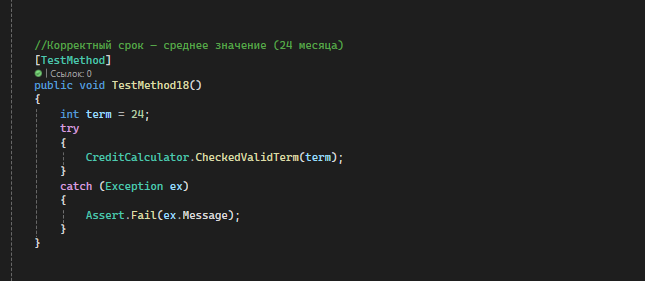
 Ожидаемый результат: выполнение метода (CheckedValidTerm(term));

Рис. 27.

Название теста: TestMethod19;

Назначение теста: Точное значение минимального срока (6 месяцев) (Метод - CheckedValidTerm(term));

Входные данные: int term = 6;

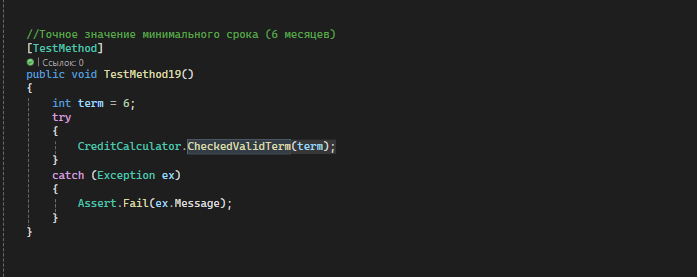
 Ожидаемый результат: выполнение метода (CheckedValidTerm(term));

Рис. 28.

Название теста: TestMethod20;

Назначение теста: Точное значение максимального срока (360 месяцев) (Метод - CheckedValidTerm(term));

Входные данные: int term = 360;

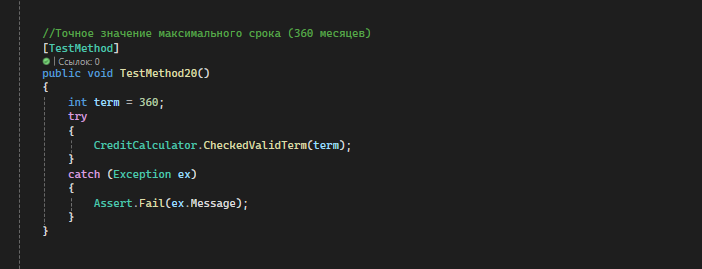
 Ожидаемый результат: выполнение метода (CheckedValidTerm(term));

Рис. 29.

## **1.6 Используемые инструменты**

Используемый язык программирования для работы: C#;

среда разработки: приложение Xamarin Forms;

## **1.7 Описание пользовательского интерфейса**

WelcomePage.xaml состоит из:

* текстовое поле (Label) – «Welcome»;
* поле для ввода данных (Entry) – username;
* поле для ввода данных (Entry) – password;
* кнопка (Button) – «Sign in»;
* поле для обозначения наличия (CheckBox) - rem;
* поле для ввода данных (Entry) – «Remember me»;
* поле для ввода данных (Entry) – «I forgot!»

Page1.xaml состоит из:

* текстовое поле (Label) – «Сумма кредита: »;
* поле для ввода данных (Entry) – CreditSumEntry;
* текстовое поле (Label) – «Срок (месяцев): »;
* поле для ввода данных (Entry) – TermEntry;
* текстовое поле (Label) – «Вид платежа: »;
* поле для выбора данных (Picker) – PaymentTypePicker;
* текстовое поле (Label) – «Процентная ставка: »;
* сайдер (Slider) – PercentSlider;
* процентное поле (Label) – PercentLabel;
* кнопка (Button) – «Рассчитать»;
* текстовое поле (Label) – «Ежемесячный платеж: »;
* текстовое поле (Label) – «Общая сумма: »;
* текстовое поле (Label) – «Переплата: »;

Page2.xaml состоит из:

* текстовое поле (Label) – «Центробанк РФ: »;
* текстовое поле (Label) – «USD: »;
* текстовое поле (Label) – «79.000»;
* текстовое поле (Label) – «EUR: »;
* текстовое поле (Label) – «90.010»;
* текстовое поле (Label) – «Текущая дата: »;
* Поле для выбора даты (DatePicker) – CurrencyDatePicker;

## **1.8 Приложение**

WelcomePage.xaml

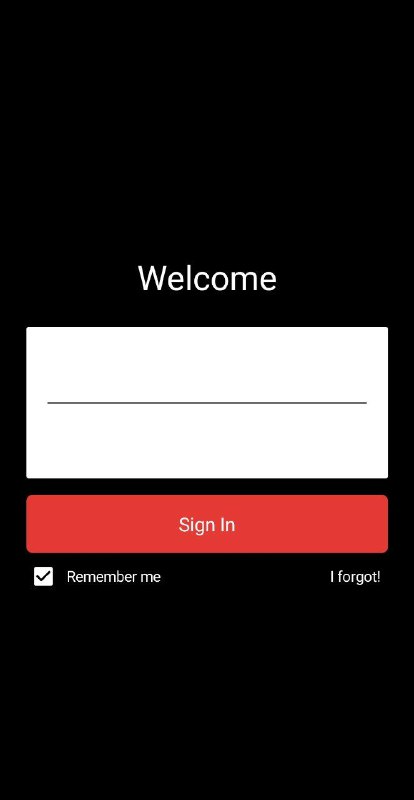


Рис. 30.

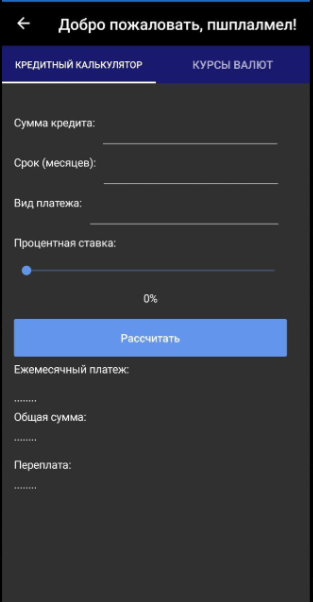
 Page1.xaml

Рис. 31.

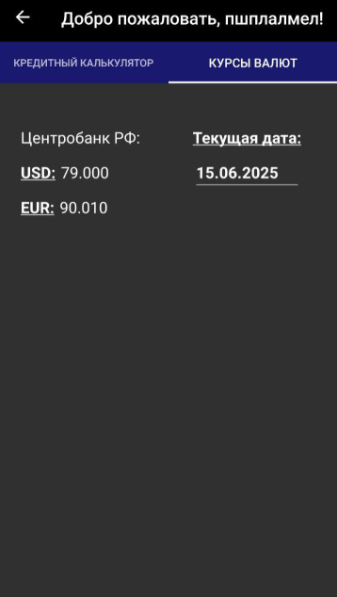


Рис. 32.