Министерство науки и высшего образования Российской Федерации
Калужский филиал
федерального государственного бюджетного
образовательного учреждения высшего образования
«Московский государственный технический университет имени Н.Э. Баумана (национальный исследовательский университет)»
(КФ МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ ***ИУК «Информатика и управление»***

КАФЕДРА ***ИУК4 «Программное обеспечение ЭВМ, информационные технологии»***

ЛАБОРАТОРНАЯ РАБОТА №4

«Разработка мобильного приложения с использованием инструмента Xamarin»

ДИСЦИПЛИНА: «Кроссплатформенная разработка ПО»

Выполнил: студент гр. ИУК4-62Б _____ (Карельский М.К.)
(Подпись)

Проверил: _____ (Пчелинцева Н.И.)
(Подпись)

Дата сдачи (защиты):

Результаты сдачи (защиты):

- Балльная оценка:
- Оценка:

Калуга, 2023

Цель: разработка кроссплатформенного мобильного приложения с использованием инструмента Xamarin.

Задачи:

1. Изучить возможности и принципы работы с Xamarin.
2. Получить навыки разработки кроссплатформенных мобильных приложений с использованием Xamarin.

Задание:

Разработать кроссплатформенное мобильное приложение согласно варианту. Для создания приложения использовать инструмент Xamarin.

Вариант 8

Таблица истинности

В приложении должна быть возможность ввода исходных данных (логическое выражение) и отображение результата (таблица).

Листинг:

MainPage.xaml.cs

```
using System;
using System.Collections.Generic;
using System.Linq;
using Xamarin.Forms;

namespace LW4
{
    public partial class MainPage : ContentPage
    {
        private Dictionary<string, int> operatorPriorities = new
Dictionary<string, int>()
        {
            { "*", 3 },
            { "+", 2 },
            { "(", 1 },
            { "!(", 1 }
        };

        private List<string> postfixList = new List<string>();
        private List<string> letters = new List<string>();
        private Dictionary<string, int> values = new Dictionary<string, int>();

        private string possibleLetters = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

        public MainPage()
        {
            InitializeComponent();
        }

        private void OnSolveButtonClicked(object sender, EventArgs e)
        {
            resultGrid.Children.Clear();
            resultGrid.RowDefinitions.Clear();
            resultGrid.ColumnDefinitions.Clear();

            Stack<string> operatorStack = new Stack<string>();
```

```
postfixList.Clear();
letters.Clear();
values.Clear();

List<string> tokenList = inputEntry.Text.Split().ToList();

for (int i = 0; i < tokenList.Count; ++i)
{
    string token = tokenList[i];
    if (possibleLetters.Contains(token))
    {
        if (!letters.Contains(token))
            letters.Add(token);
        postfixList.Add(token);
    }
    else if (token == "!")
    {
        if (possibleLetters.Contains(tokenList[i + 1]))
        {
            if (!letters.Contains(tokenList[i + 1]))
                letters.Add(tokenList[i + 1]);
            postfixList.Add(tokenList[i + 1]);
            postfixList.Add("!");
        }
        else
            operatorStack.Push("!(");
        ++i;
    }
    else if (token == "(")
        operatorStack.Push(token);
    else if (token == ")")
    {
        string topToken = operatorStack.Pop();
        while (topToken != "(" && topToken != "!(")
        {
            string nextToken = operatorStack.Pop();
            postfixList.Add(topToken);
            if (nextToken == "!(")
                postfixList.Add("!");
            topToken = nextToken;
        }
    }
    else
    {
        while (operatorStack.Count > 0 &&
            operatorPriorities[operatorStack.Peek()] >=
            operatorPriorities[token])
            postfixList.Add(operatorStack.Pop());
        operatorStack.Push(token);
    }
}

while (operatorStack.Count > 0)
    postfixList.Add(operatorStack.Pop());

resultGrid.RowDefinitions.Add(new RowDefinition());
foreach (string letter in letters)
{
    values.Add(letter, 0);

    resultGrid.ColumnDefinitions.Add(new ColumnDefinition());
    resultGrid.Children.Add(new Label() { Text = letter },
        resultGrid.ColumnDefinitions.Count - 1, 0);
}
resultGrid.ColumnDefinitions.Add(new ColumnDefinition());
resultGrid.Children.Add(new Label() { Text = "Result" },
```

```
                    resultGrid.ColumnDefinitions.Count - 1, 0);

            for (int i = 0; i < Math.Pow(2, letters.Count); ++i)
            {
                int number = i;
                resultGrid.RowDefinitions.Add(new RowDefinition());
                foreach (string letter in letters)
                {
                    values[letter] = number % 2;
                    number /= 2;

                    resultGrid.Children.Add(new Label() { Text =
values[letter].ToString() },
                        letters.IndexOf(letter), resultGrid.RowDefinitions.Count
- 1);
                }

                int result = Convert.ToInt32(Solve(out int _));
                resultGrid.Children.Add(new Label() { Text = result.ToString()
},
                        resultGrid.ColumnDefinitions.Count - 1,
resultGrid.RowDefinitions.Count - 1);
            }
        }

        private bool Solve(out int nextIndex, int index=-1, bool negation=false)
        {
            if (index == -1)
                index = postfixList.Count - 1;

            if (postfixList[index] == "+" || postfixList[index] == "*")
            {
                bool second = Solve(out int indexForFirst, index - 1);
                bool first = Solve(out nextIndex, indexForFirst);
                if (postfixList[index] == "+")
                    return negation ^ (first || second);
                return negation ^ (first && second);
            }
            if (postfixList[index] == "!")
                return Solve(out nextIndex, index - 1, true);
            nextIndex = index - 1;
            return negation ^ Convert.ToBoolean(values[postfixList[index]]);
        }
    }
}
```

MainPage.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
             xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
             x:Class="LW4.MainPage">
    <StackLayout>
        <Label Padding="20,20,0,0" Text="Введите логическую формулу" />
        <Label Padding="20,0,0,0" Text="Возможные элементы (разделять
пробелами):" />
        <Label Padding="30,0,0,0" Text="- Переменные от A до Z" />
        <Label Padding="30,0,0,0" Text="- Знаки '+', '*', '!', '(', ')'" />

        <Entry
            x:Name="inputEntry"
            TextTransform="Uppercase"
            Margin="10,0,10,0"
            Placeholder="A + ( B * ( C + D ) ) + ! ( E * F * ! G )" />
```

```
        <Button
            x:Name="solveButton"
            Text="Решить"
            Clicked="OnSolveButtonClicked" />

        <ScrollView Padding="10,0,10,0">
            <Grid x:Name="resultGrid"/>
        </ScrollView>
    </StackLayout>
</ContentPage>
```

Результат:

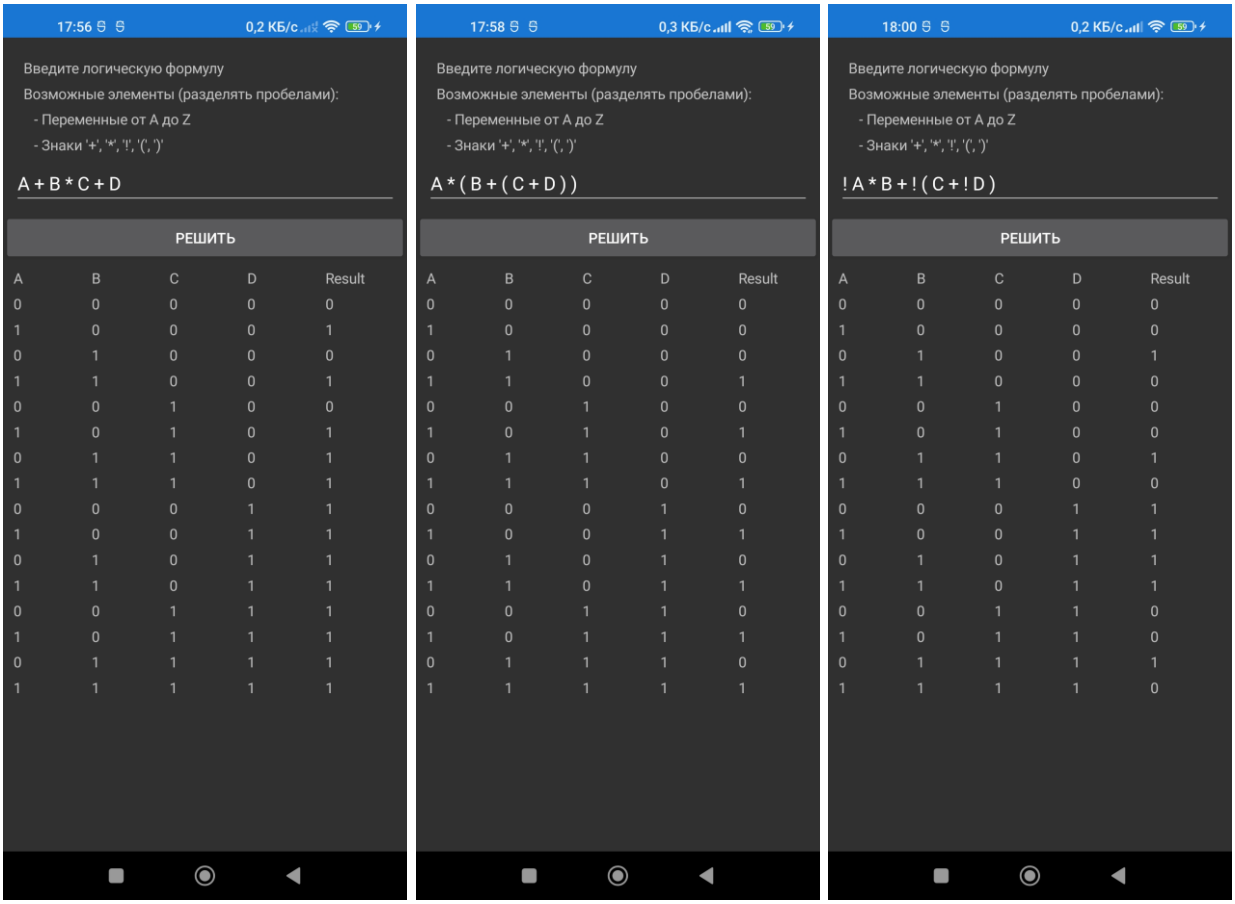

Рис. 1. Результат

Вывод: в ходе выполнения лабораторной работы были получены практические навыки разработки кроссплатформенного мобильного приложения с использованием инструмента Xamarin.