

Київський національний університет
імені Т.Шевченка

Звіт

до лабораторної роботи №1
на тему:

«Реалізація арифметичного калькулятора»

**Студента 4-го курсу
Бурлаки Владислава
Варіант №15**

Київ-2023

Зміст

Постановка задачі.....	3
Короткий виклад теорії.....	4
Опис алгоритмів, які застосовуються для обчислень.....	5
Опис та демонстрація роботи програми.....	5
Список використаних джерел.....	7
Додаток (код основної частини програми).....	7

Постановка задачі

Метою роботи є формування базових знань та умінь розробляти для мобільного пристрою програми обчислювального характеру, використовувати програмні засоби роботи з файлами та візуалізувати дані. За бажанням студента погодженням з боку викладача, зміст роботи може бути змінений.

Загальні вимоги:

- Має бути документ (PDF) з описом вимог до програмного продукту (ТЗ)
- Програма повинна реалізовувати запис/читання даних з файлу або локальної БД
- Програма повинна мати пункт з інформацією про розробника та його/її фото
- Повинні бути застосовані базові принципи UX-дизайну та ергономіки для мобільних пристроїв

Варіант №15:

Обирається система числення і вводяться значення операндів. Значення виконання операції та результат конвертування числа в інші числові системи. Передбачене збереження у файлі введених значень. Операції: + - * /; крім 10-ї, взяти системи числення з основою 2, 8, 16; реалізувати взаємний конвертор.

Короткий виклад теорії

Арифметичний калькулятор для мобільних платформ є універсальним інструментом, який враховує різноманітність числових систем. Незалежно від того, чи вам потрібно виконати обчислення в двійковій, восьмеричній, десятковій чи шістнадцятковій системах числення, цей додаток забезпечить вас зручним та ефективним інструментом для вирішення завдань у будь-якому форматі.

Мобільний арифметичний калькулятор призначений для використання в повсякденному житті, де користувачам може знадобитися провести швидкі математичні обчислення в різних сценаріях. Він пропонує інтуїтивний інтерфейс, що спрощує його використання.

Здатність працювати з різними числовими системами розширює функціонал калькулятора, надаючи користувачам можливість вибору оптимального формату для їхніх завдань. Це особливо важливо в тих випадках, коли вам потрібно перевести значення між різними системами числення або виконати операції, які вимагають конкретного формату введення або виведення.

Цей додаток може стати надійним помічником для різних категорій користувачів, від студентів і вчителів до працівників і простих користувачів, які шукають простий і зручний спосіб вирішувати математичні завдання у мобільному форматі. Його практичність і доступність роблять його важливим інструментом в цифровому повсякденні.

Опис алгоритмів, які застосовуються для обчислень

Цей застосунок є арифметичним калькулятором для виконання різноманітних операцій над двома операндами в різних числових системах. Після введення усіх необхідних даних та вибору арифметичної операції додаток зчитує їх, конвертує в єдину систему числення та проводить обрахунок результату.

Значення результату арифметичних розрахунків зберігається у відповідній змінній. У разі вибору користувачем нової системи числення результат буде відповідно конвертовано та відображено.

Опис та демонстрація роботи програми

Цей застосунок є арифметичним калькулятором для виконання різноманітних операцій над двома операндами в різних числових системах. Користувачу надана можливість обрати систему запису вхідних даних, арифметичну операцію та ввести два операнди. Після цього ви можете натиснути кнопку "Calculate" та отримати результат арифметичних операцій (рис. 1). Також вам доступна конвертація результату в одну з запропонованих числових систем. Ви також можете зберігати та завантажувати дані про свої розрахунки, використовуючи кнопки "Save to File" та "Load from File".

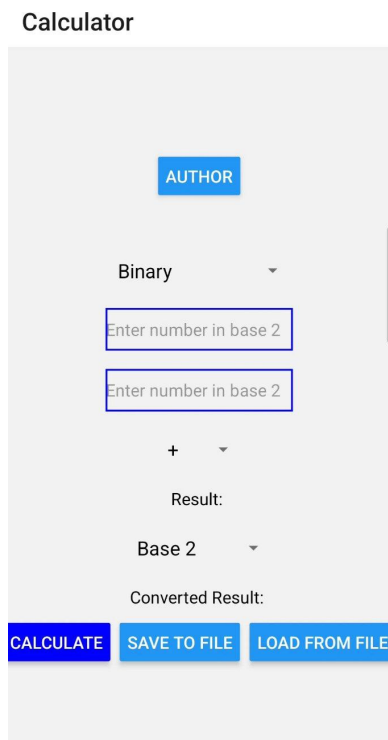


Рисунок 1 - головна сторінка застосунку

Ознайомитись з автором застосунку Ви можете після натискання кнопки “Author”. Повернутись до головної сторінки застосунку допоможе кнопка “Calc” (рис. 2).

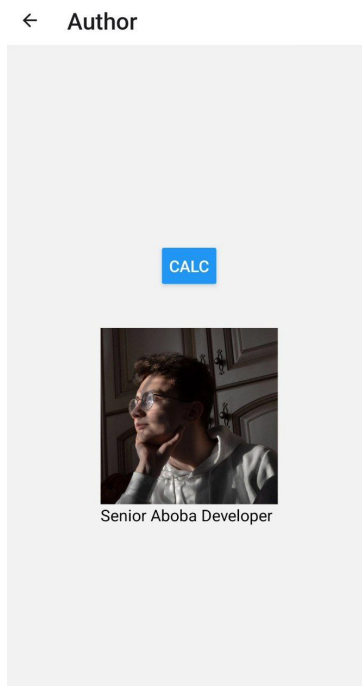


Рисунок 2 - сторінка з автором

Список використаних джерел

1. Офіційна документація. *Expo*. URL: <https://expo.dev/>
2. Офіційна Документація react native. URL: <https://reactnative.dev/>

Додаток (код основної частини програми)

```
import { useNavigation } from '@react-navigation/native';
import React from 'react';
import { Button } from 'react-native';
import { Text } from 'react-native';
import { View, Image } from 'react-native';

const AuthorScreen = () => {

  const navigation = useNavigation();
  const goToCalk = () => {
    navigation.navigate("Calculator");
  };

  return (
    <View style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>
      <View style={{paddingBottom: 50}}>
        <Button title='Calc' onPress={goToCalk}/>
      </View>
      <View>
        <Image source={require('../assets/ava.jpg')}
          style={{width: 200, height: 200}} />
        <Text>Senior Aboba Developer</Text>
      </View>
    </View>
  );
};

export default AuthorScreen;
```

```
import { View, Text, TextInput, Button } from 'react-native';
import * as FileSystem from 'expo-file-system';
```

```
import { Picker } from '@react-native-picker/picker';
import { useNavigation } from '@react-navigation/native';
import { useState } from 'react';

const Calculator = () => {
  const [number1, setNumber1] = useState('');
  const [number2, setNumber2] = useState('');
  const [selectedBase, setSelectedBase] = useState('2');
  const [selectedOperation, setSelectedOperation] = useState('+');
  const [result, setResult] = useState('');
  const [selectedResultBase, setSelectedResultBase] = useState('2');
  const navigation = useNavigation();

  const calculateResult = () => {
    let num1 = parseInt(number1, parseInt(selectedBase));
    let num2 = parseInt(number2, parseInt(selectedBase));
    let res = 0;
    switch (selectedOperation) {
      case '+':
        res = num1 + num2;
        break;
      case '-':
        res = num1 - num2;
        break;
      case '*':
        res = num1 * num2;
        break;
      case '/':
        res = num1 / num2;
        break;
      case '^':
        res = num1 ** num2;
        break;
      default:
        break;
    }
    setResult(res.toString());
  };

  const convertBase = (number) => {
    return parseInt(number, 10).toString(parseInt(selectedResultBase));
  };

  const bases = ["2", "8", "10", "16"];
```



```

const saveToFile = async () => {
  try {
    const content = {
      number1,
      number2,
      selectedBase,
      selectedOperation,
      result,
      selectedResultBase,
    };
    const fileUri = `${FileSystem.documentDirectory}/calculator_data.json`;
    await FileSystem.writeAsStringAsync(fileUri, JSON.stringify(content));
    console.log('Data saved to file.');
```

```

  } catch (error) {
    console.error('Error saving data:', error);
  }
};

const loadFromFile = async () => {
  try {
    const fileUri = `${FileSystem.documentDirectory}/calculator_data.json`;
    const content = await FileSystem.readAsStringAsync(fileUri);
    const data = JSON.parse(content);
    setNumber1(data.number1);
    setNumber2(data.number2);
    setSelectedBase(data.selectedBase);
    setSelectedOperation(data.selectedOperation);
    setResult(data.result);
    setSelectedResultBase(data.selectedResultBase);
    console.log('Data loaded from file.');
```

```

  } catch (error) {
    console.error('Error loading data:', error);
  }
};

const goToAuthor = () => {
  navigation.navigate('Author');
};

return (
  <View style={{ flex: 1, justifyContent: 'center', alignItems: 'center' }}>
    <View style={{paddingBottom: 50}}>
      <Button title='Author' onPress={goToAuthor}/>
    </View>
  </View>
);

```

```

<Picker
  selectedValue={selectedBase}
  style={{ height: 50, width: "50%" }}
  onChange={(itemValue) => setSelectedBase(itemValue)}
>
  <Picker.Item label="Binary" value="2" />
  <Picker.Item label="Octal" value="8" />
  <Picker.Item label="Decimal" value="10" />
  <Picker.Item label="Hexadecimal" value="16" />
</Picker>
<TextInput
  style={{ height: 45, borderColor: 'blue', borderWidth: 2, margin: 10,
width: 200 }}
  onChangeText={(text) => setNumber1(text)}
  value={number1}
  placeholder={`Enter number in base ${selectedBase}`}
/>
<TextInput
  style={{ height: 45, borderColor: 'blue', borderWidth: 2, margin: 10,
width: 200 }}
  onChangeText={(text) => setNumber2(text)}
  value={number2}
  placeholder={`Enter number in base ${selectedBase}`}
/>
<Picker
  selectedValue={selectedOperation}
  style={{ height: 50, width: 100 }}
  onChange={(itemValue) => setSelectedOperation(itemValue)}
>
  <Picker.Item label="+" value="+" />
  <Picker.Item label="-" value="-" />
  <Picker.Item label="*" value="*" />
  <Picker.Item label="/" value="/" />
  <Picker.Item label="^" value="^" />
</Picker>
<Text style={{ margin: 10 }}>Result: {result}</Text>
<Picker
  selectedValue={selectedResultBase}
  style={{ height: 50, width: "40%" }}
  onChange={(itemValue) => setSelectedResultBase(itemValue)}
>
  {bases.map((base) => (
    <Picker.Item key={base} label={`Base ${base}`} value={base} />
  ))}
</Picker>

```

```
<Text style={{ margin: 10 }}>
  Converted Result: {result ? convertBase(result) : ''}
</Text >
<View style={{display:'flex', flexDirection:'row'}}>
  <View style={{ margin:5}}>
    <Button color="#0000ff" title="Calculate" onPress={calculateResult} />
  </View>
  <View style={{ margin:5}}>
    <Button title="Save to File" onPress={saveToFile} />
  </View>
  <View style={{ margin:5}}>
    <Button title="Load from File" onPress={loadFromFile} />
  </View>
</View>
);
};

export default Calculator;
```