Міністерство освіти і науки України
Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»
Факультет інформатики та обчислювальної техніки
Кафедра обчислювальної техніки

Розрахунково-графічна робота

з дисципліни

«Об’єктно орієнтоване програмування»

Виконав:
Студент групи ІМ-31
Максимовський Назар Русланович
номер у списку групи: 13

Перевірив:
Порєв В.М.

Київ 2023

Завдання

1. Створити програму, яка зберігає список студентів групи і виконує розрахунки успішності навчання
2. Скомпілювати проєкт і отримати виконуваний файл програми.
3. Перевірити роботу програми. Налагодити програму.
4. Проаналізувати та прокоментувати результати та вихідний текст програми.
5. Оформити звіт.

Обґрунтування проектного рішення

Метою проєкту є створення програми для збереження списку студентів групи та розрахунку їхньої успішності. Kotlin обрано через його зручний синтаксис і можливості для роботи з даними, а Gradle — для автоматизації збірки та керування залежностями.

Аналіз можливих варіантів

Розглядалися варіанти використання Java та Kotlin. Kotlin обрано через його сучасність, компактний код і зручність обробки даних. Це дозволяє швидко реалізовувати функції з розрахунків і збереження інформації.

Теоретичні положення

Проєкт базується на принципах об'єктно-орієнтованого програмування для модульності та повторного використання коду. Kotlin забезпечує надійність типізації та функціональні можливості, необхідні для розрахунків.

Опис реалізації

Програма включає базу студентів групи, яка зберігається у вигляді структурованих даних. Передбачено функціонал для введення, перегляду, редагування списку студентів і розрахунку середнього балу. Основні етапи: створення структури даних, реалізація функцій для обробки та аналізу успішності, тестування.

Вихідні тексти файлів програми

Group.kt

```kotlin
package com.example.rgr

class Group(val students: List<Student>) {
    fun calculateAverageGrade(studentName: String): Double? {
        val student = students.find { it.firstName == studentName ||
it.lastName == studentName }
        return student?.grades?.average()
    }

    fun calculateGroupAverage(): Double {
        val allGrades = students.flatMap { it.grades }
        return if (allGrades.isNotEmpty()) {
            allGrades.average()
        } else {
            0.0
        }
    }
}
```

MainActivity.kt

```kotlin
package com.example.rgr

import android.os.Bundle
import android.view.View
import android.widget.Button
import android.widget.EditText
import android.widget.TextView
import androidx.appcompat.app.AppCompatActivity
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView

class MainActivity : AppCompatActivity() {

    private val students = mutableListOf<Student>()
    private lateinit var recyclerView: RecyclerView
    private lateinit var adapter: StudentAdapter

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_main)

        val editTextName = findViewById<EditText>(R.id.editTextName)
        val editTextGrades = findViewById<EditText>(R.id.editTextGrades)
        val buttonAddStudent = findViewById<Button>(R.id.buttonAddStudent)
        val buttonCalculate = findViewById<Button>(R.id.buttonCalculate)
        val textViewResult = findViewById<TextView>(R.id.textViewResult)
        val editTextLastName = findViewById<EditText>(R.id.editTextLastName)


        buttonCalculate.setOnClickListener {
            val group = Group(students)
            val groupAverage = group.calculateGroupAverage()
            textViewResult.text = "Середній бал групи: $groupAverage"
        }

        recyclerView = findViewById(R.id.recyclerViewStudents)
        adapter = StudentAdapter(students)
        recyclerView.adapter = adapter
        recyclerView.layoutManager = LinearLayoutManager(this)

        buttonAddStudent.setOnClickListener {
```

```
            val firstName = editTextName.text.toString()
            val lastName = editTextLastName.text.toString()
            val grades = editTextGrades.text.toString().split(",").map {
it.toDouble() }
            val student = Student(firstName, lastName, grades)
            students.add(student)

            editTextName.text.clear()
            editTextLastName.text.clear()
            editTextGrades.text.clear()

            // Сортування та оновлення RecyclerView після додавання нового
студента
            students.sortBy { it.lastName }
            updateRecyclerView()
        }
    }

    private fun updateRecyclerView() {
        adapter.notifyDataSetChanged()
    }
}
```

Student.kt

```
package com.example.rgr

data class Student(val firstName: String, val lastName: String, val grades:
List<Double>)
```

StudentAdapter.kt

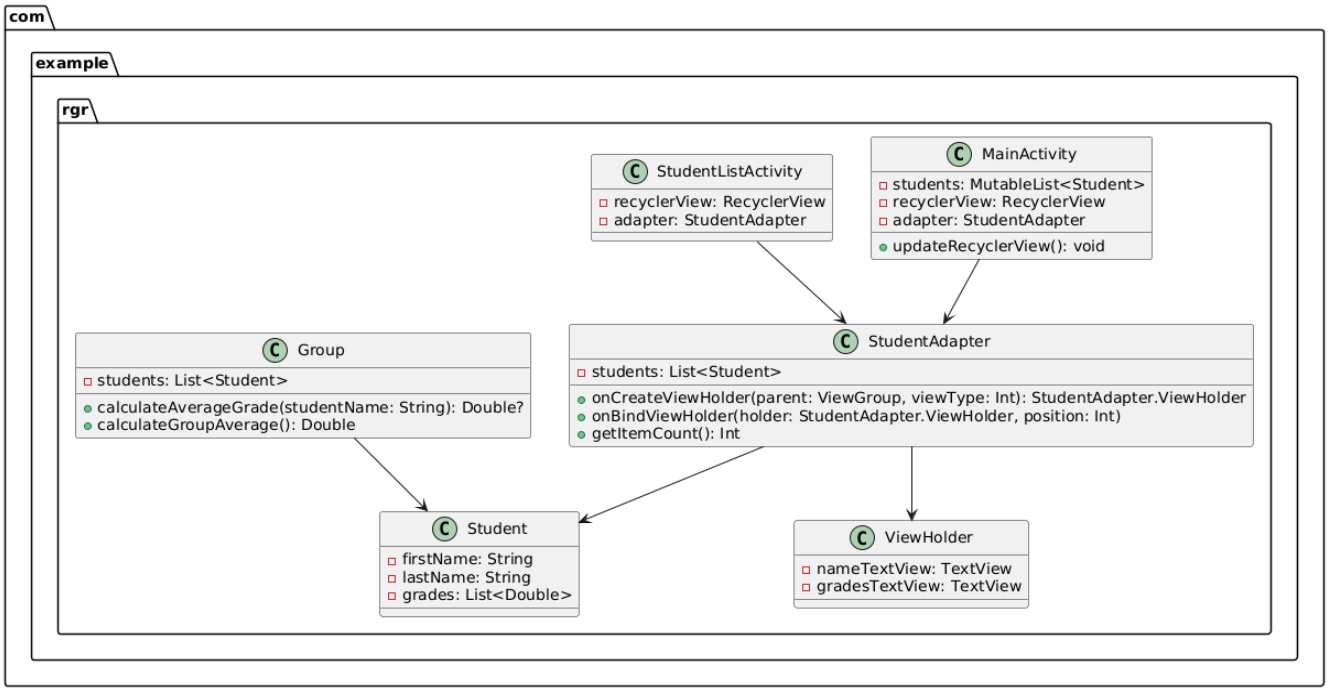
```
package com.example.rgr

import android.view.LayoutInflater
import android.view.View
import android.view.ViewGroup
import android.widget.TextView
import androidx.recyclerview.widget.RecyclerView

class StudentAdapter(private val students: List<Student>) :
RecyclerView.Adapter<StudentAdapter.ViewHolder>() {

    class ViewHolder(itemView: View) : RecyclerView.ViewHolder(itemView) {
        val nameTextView: TextView = itemView.findViewById(R.id.textViewName)
        val gradesTextView: TextView =
itemView.findViewById(R.id.textViewGrades)
    }

    override fun onCreateViewHolder(parent: ViewGroup, viewType: Int):
ViewHolder {
        val view =
LayoutInflater.from(parent.context).inflate(R.layout.student_item, parent,
false)
        return ViewHolder(view)
    }

    override fun onBindViewHolder(holder: ViewHolder, position: Int) {
        val student = students[position]
```

```
        holder.nameTextView.text = "${student.lastName} ${student.firstName}"
        holder.gradesTextView.text = student.grades.joinToString(", ")
    }

    override fun getItemCount(): Int {
        return students.size
    }
}
```

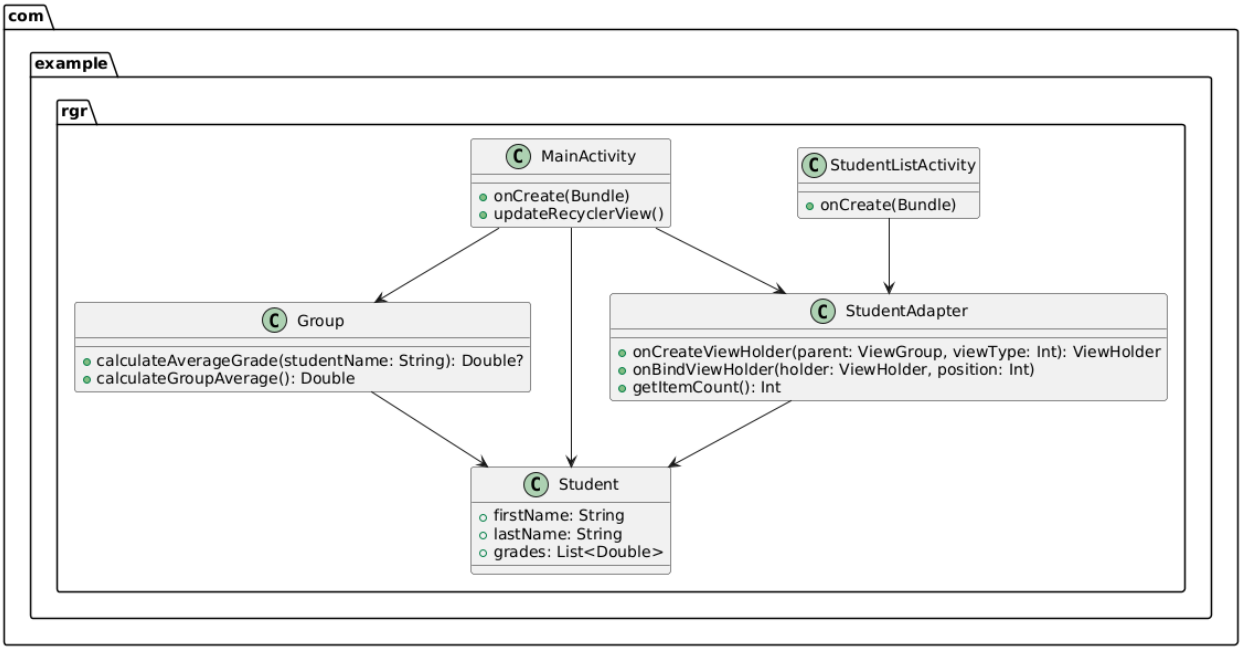
StudentListActivity.kt

```
package com.example.rgr

import android.os.Bundle
import androidx.appcompat.app.AppCompatActivity
import androidx.recyclerview.widget.LinearLayoutManager
import androidx.recyclerview.widget.RecyclerView

class StudentListActivity : AppCompatActivity() {

    private lateinit var recyclerView: RecyclerView
    private lateinit var adapter: StudentAdapter

    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContentView(R.layout.activity_student_list)

        val students = intent.getSerializableExtra("students") as
List<Student>

        recyclerView = findViewById(R.id.recyclerViewStudents)
        adapter = StudentAdapter(students)
        recyclerView.adapter = adapter
        recyclerView.layoutManager = LinearLayoutManager(this)
    }
}
```

Діаграма класів програми



Діаграма залежностей між модулями та пакетами

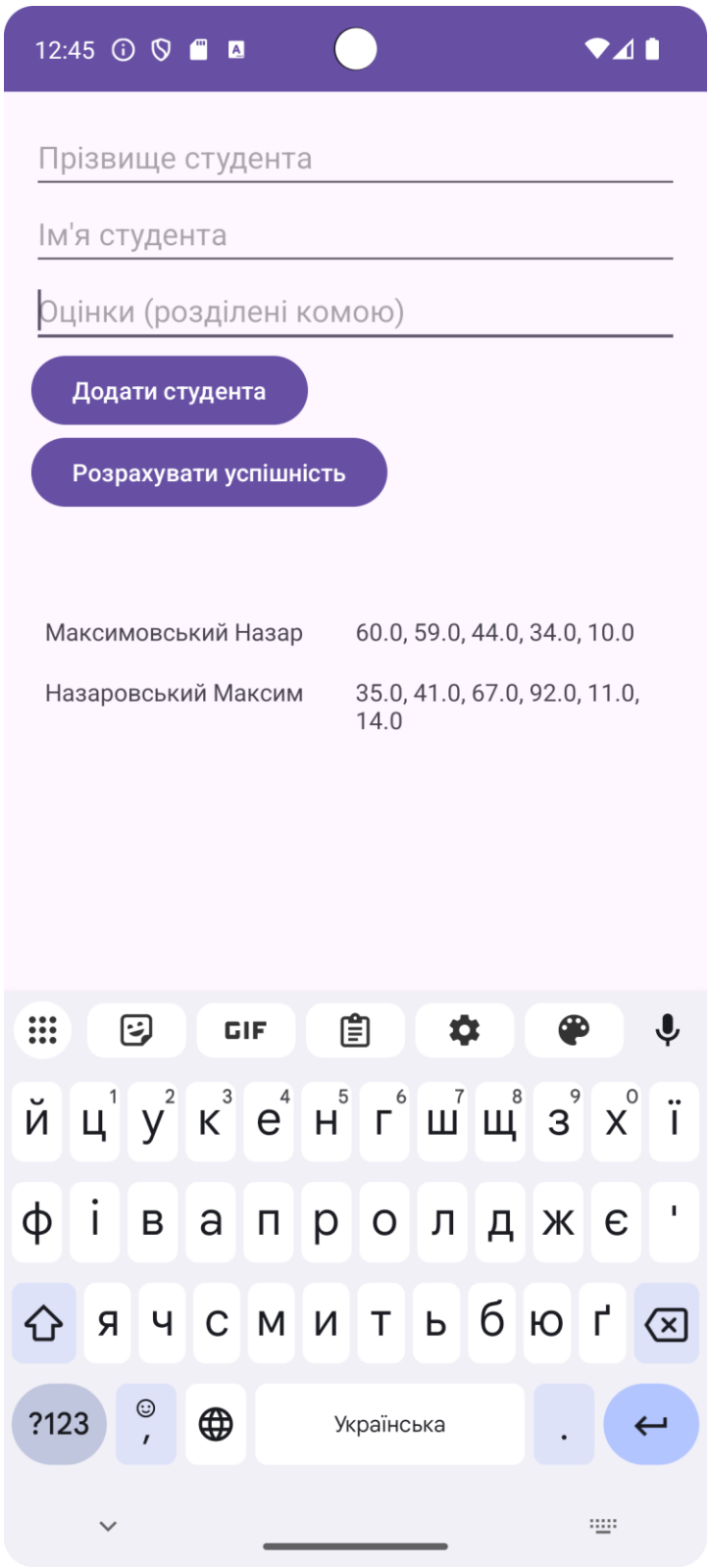


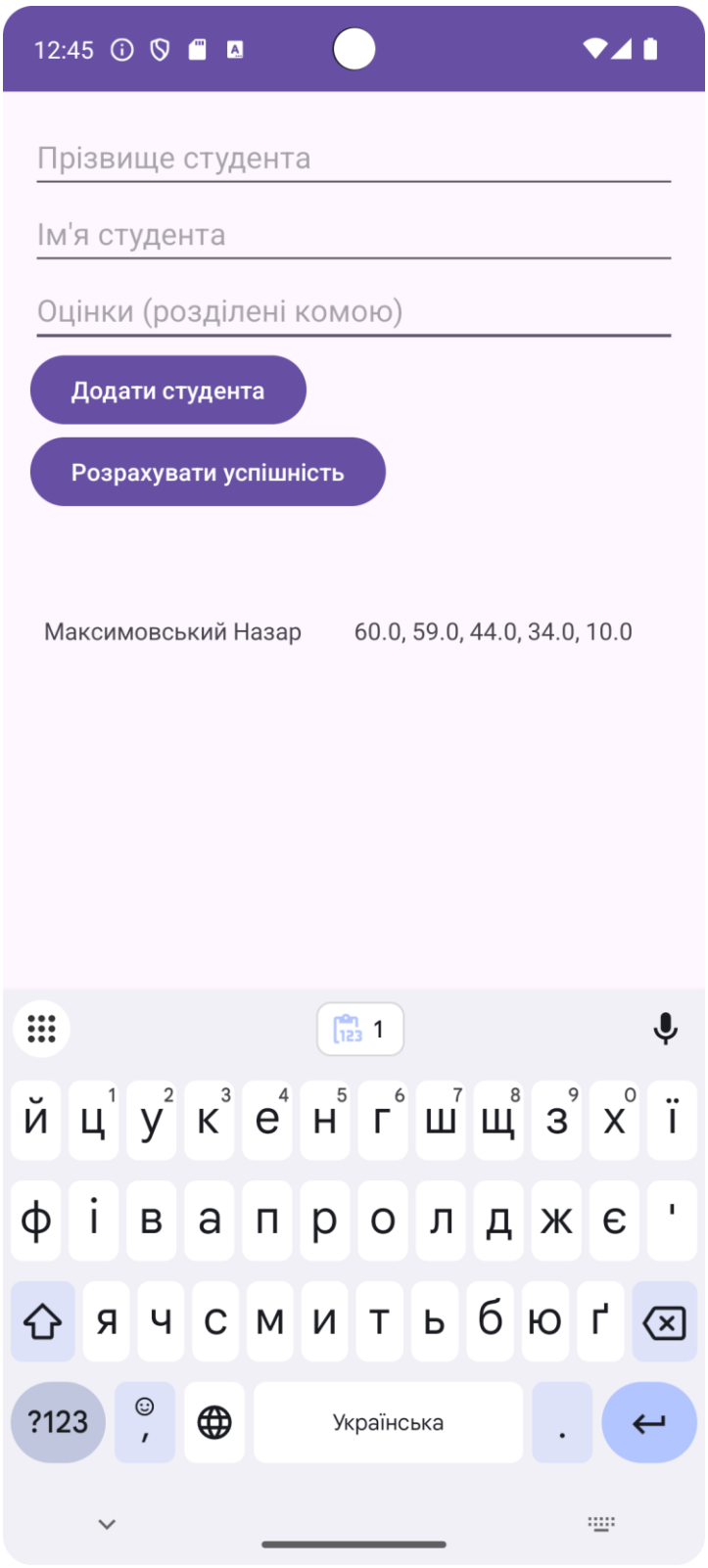
Позначення для модифікаторів видимості

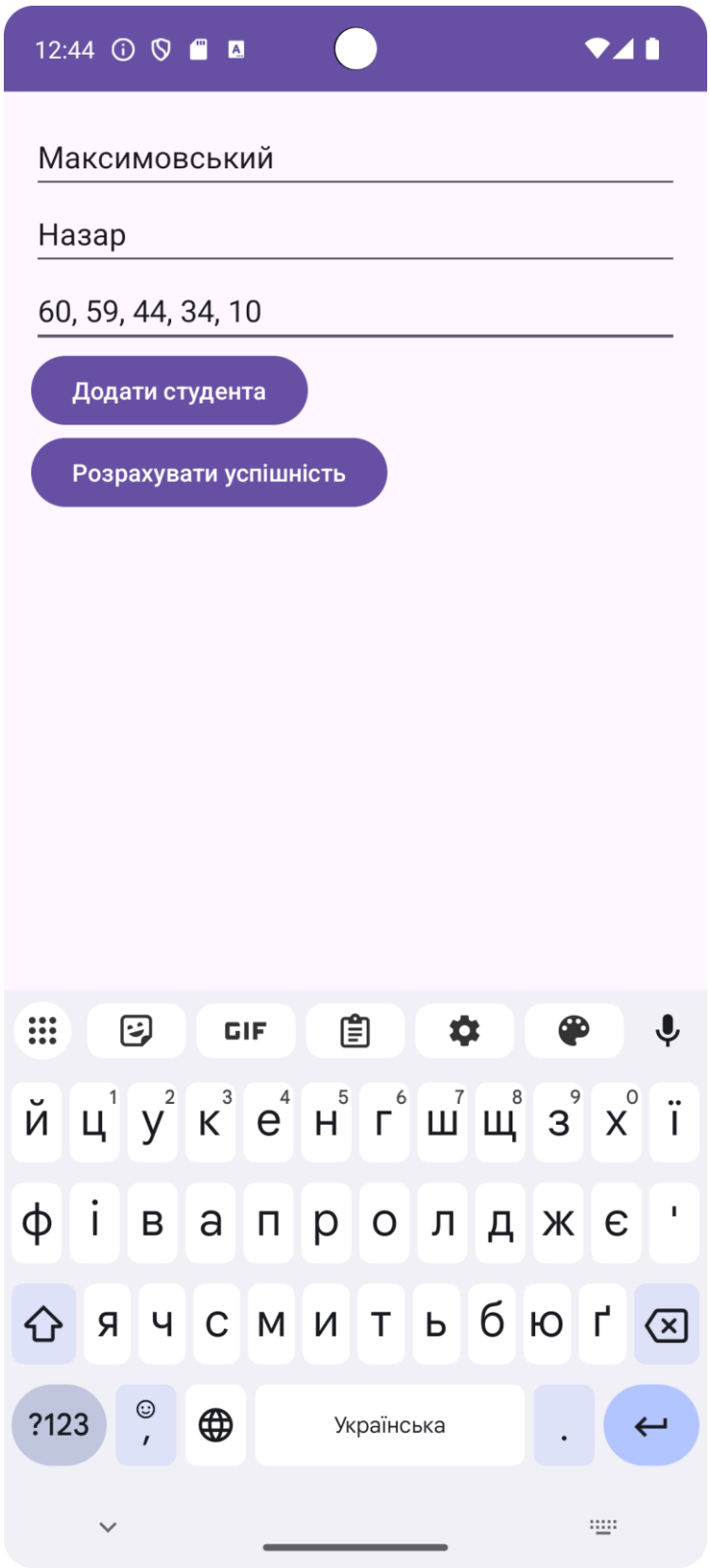
Позначення для поля	*Позначення для методу*	*Модифікатор видимості*
□	■	`private`
◇	◆	`protected`

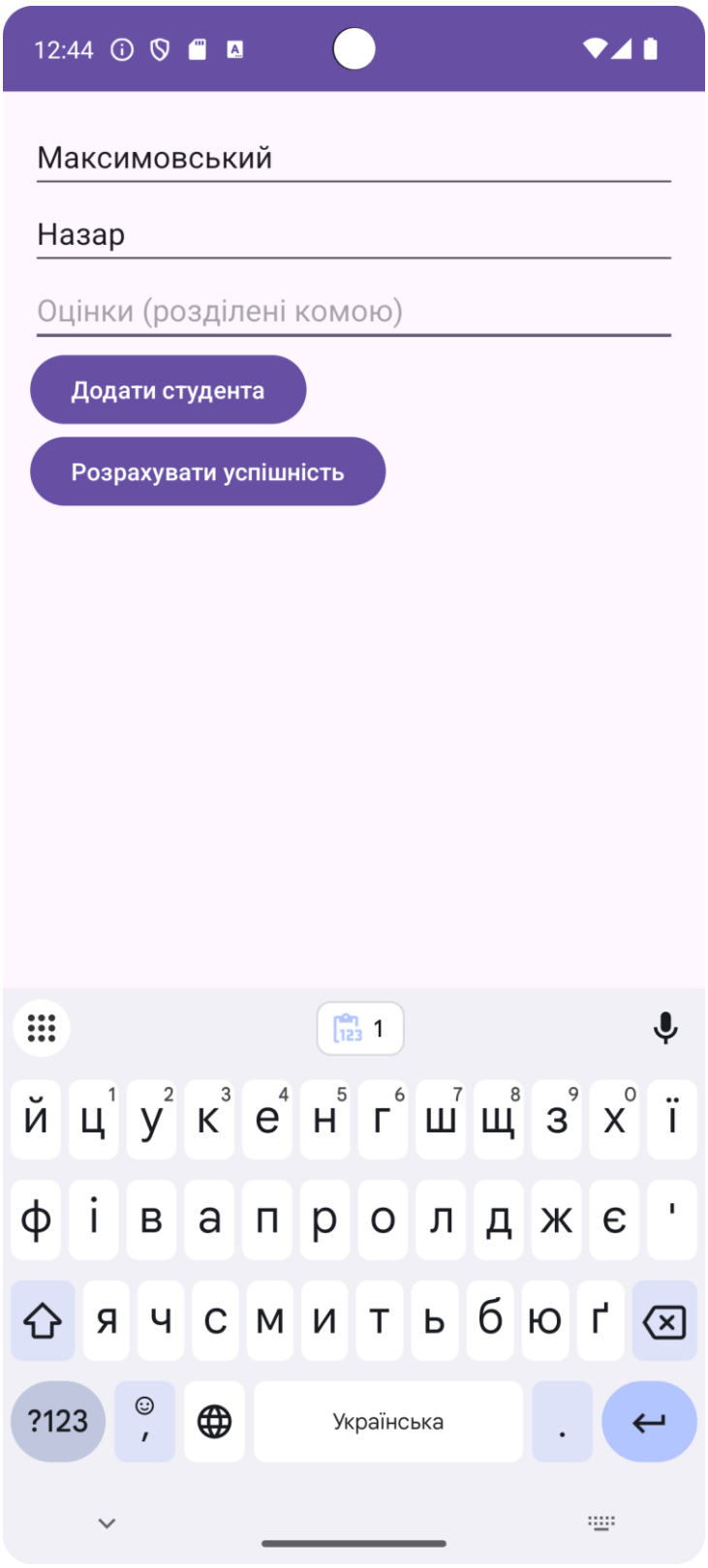
○	●	public

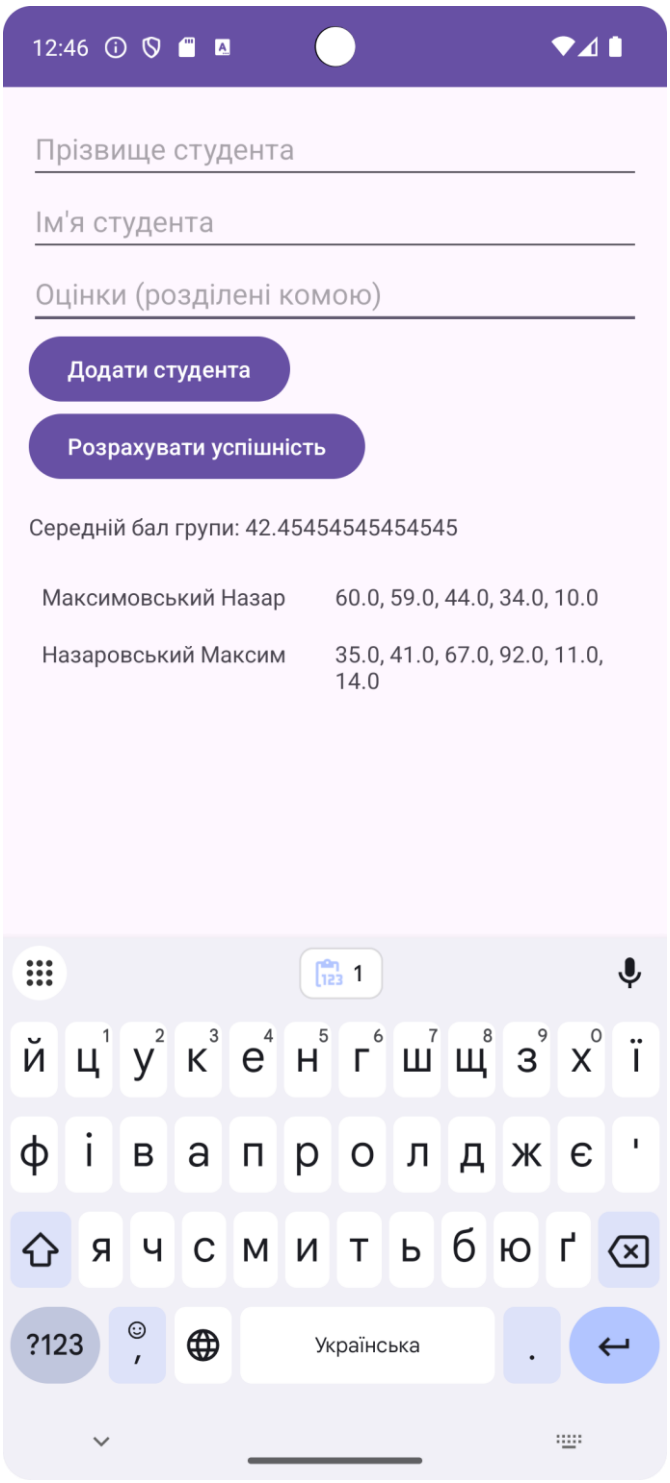
Ілюстрації виконання програми


12:45
Прізвище студента
Ім'я студента
Оцінки (розділені комою)
Додати студента
Розрахувати успішність
Максимовський Назар
60.0, 59.0, 44.0, 34.0, 10.0
Назаровський Максим
35.0, 41.0, 67.0, 92.0, 11.0, 14.0
GIF
Українська


12:45
Прізвище студента
Ім'я студента
Оцінки (розділені комою)
Додати студента
Розрахувати успішність
Максимовський Назар 60.0, 59.0, 44.0, 34.0, 10.0
й ц у к е н г ш щ з х ї
ф і в а п р о л д ж є '
я ч с м и т ь б ю ґ
?123
Українська


12:44
Максимовський
Назар
60, 59, 44, 34, 10
Додати студента
Розрахувати успішність
GIF
Українська


12:44
Максимовський
Назар
Оцінки (розділені комою)
Додати студента
Розрахувати успішність
1
й ц у к е н г ш щ з х ї
ф і в а п р о л д ж є '
я ч с м и т ь б ю ґ
?123
Українська



Висновки

Програма забезпечує збереження даних про студентів та автоматизацію розрахунків їхньої успішності. Використані технології роблять проєкт гнучким для майбутніх оновлень і розширення функціоналу.