



ASSIGNMENT 05

REQUIREMENTS

- You will be given one of the problems below to solve using feature-driven development.
- The program must provide a menu-driven console user interface.
- Use classes to represent the following:
 - The domain entity (**ComplexNumber**, **Expense**, **Student**, **Book**)
 - A **UI** class, that implements the user interface
 - A **Services** class, that implements the required functionalities
- Have 10 programmatically generated entries in the application at start-up.
- Tests and specifications for non-UI functions related to the first functionality.

1. COMPLEX NUMBERS

Manage a list of **complex numbers** in the $(a+bi)$ form and provide the user the following features:

1. Add a number to the list. The number is read from the console.
2. Show the list of numbers on the console.
3. Filter the list so that it contains only the numbers between indices p_{start} and p_{end} , where these values are read from the console.
4. Undo the last operation that modified program data. This step can be repeated.

2. EXPENSES

Manage a list of **expenses**. Each expense has a **day** (integer between 1 and 30), **amount of money** (positive integer) and **expense type** (string). Provide the user the following features:

1. Add a new expense read to the list. Expense data is read from the console.
2. Show the list of expenses on the console.
3. Filter the list so that it contains only expenses above a certain value that is read from the console.
4. Undo the last operation that modified program data. This step can be repeated.

3. STUDENTS

Manage a list of **students**. Each student has an **id** (integer, unique), a **name** (string) and a **group** (positive integer). Provide the user the following features:

1. Add a new student to the list. Student data is read from the console.
2. Show the list of students on the console.
3. Filter the list so that students in a given group (read from the console) are deleted from the list.
4. Undo the last operation that modified program data. This step can be repeated.

4. BOOKS

Manage a list of **books**. Each book has an **isbn** (string, unique), an **author** and a **title** (strings). Provide the user the following features:

1. Add a new book to the list. Book data is read from the console.
2. Show the list of books on the console.
3. Filter the list so that book titles starting with a given word are deleted from the list.
4. Undo the last operation that modified program data. This step can be repeated.