

Programming Systems and Environments, Lab 1

Jakub Grzana 241530

Task 1 - Hello World

I've created program that has contains singular package with (added in subsequent tasks) different files. Main class was changed within pom.xml file to "Program.Main" (package "Program" class Main written in "Main.java") Then I've printed welcoming message to the user, taking name of the user from environmental variable "USERNAME".

Project compiled & executed without errors, warnings nor exceptions, results were as expected.

Task 2 - Person class

I've expanded program from task 1, adding "Person.java" file with Person class. All data stored within that class was set to private, with public getters and private setters. So, for the rest of program, objects of this class are immutable, you can only get state of the object and assign new one to existing variable. Overloaded method toString() displays state of object (it's content) by using C-style string formatting. I didn't use any final fields, because it doesn't make any sense to have them within this class, maybe except for values for default constructor, but I've decided to declare this one as private, effectively deleting this constructor (this was my designer choice, there's no reason to create dummy objects of immutable class)

Project compiled & executed without errors, warnings nor exceptions, results were as expected.

Task 3 - Person class in a table

I've expanded program from task 2 by editing "Main.java" file, creating ArrayList containing objects of Person class. I've decided to use list instead of array, because in given context it makes more sense; you don't need to define upfront how many entries you want to add, all you need to do is append new objects to the end of the list, then you can iterate over this list by using range loop.

Project compiled & executed without errors, warnings nor exceptions, results were as expected.

Task 4 - Trinomial

I've expanded program from task 3 by adding "Trinomial.java" file with analogous class. This class, similarly to Person class, was designed by me to be immutable: all getters are public, all setters are private. Trinomial class solves quadratic equations and it does all calculations during creation of its object, everything is computed within constructor. This grants instant access to value of delta or zeros of given trinomial. Method "toString()" was overloaded to display neat representation of quadratic equation specified, and "Solutions()" was written to display number of zeros, and their X values. It works properly for negative value of delta (shows that there're no zeros)

Project compiled & executed without errors, warnings nor exceptions, results were as expected.

Task 5 - Complex numbers

I didn't approach this task, given there's no operator overload in java I would like to know proper way to apply Complex class to Trinomial class developed in task 4.

Conclusions

Java is, syntax-wise, very similar to C++ language. Separation into packages, importing libraries with "import" keyword and garbage collector are incredibly useful compared to counterparts in C++ (talking about #include keyword and STL classes) however lack of operator overload looks just weird. Furthermore, Java at this moment seems redundant to me: Python has much more flexible syntax, more built-in tools and for heavy computations it can use external code written in C++ I hope to see true potential of Java in future tasks.

CODE: Main.java

```
package Program;
import java.util.List;
import java.util.ArrayList;
public class Main {
    public static void main(String[] args) {
        // Hello world
        String userWindows = System.getenv("USERNAME");
        System.out.println("Hello there " + userWindows);

        // Person class task
        List<Person> personList = new ArrayList<>();
        personList.add(new Person(24, "Jakub", "Grzana"));
        personList.add(new Person(17, "Aubron", "Phivyre"));
        personList.add(new Person(28, "Elred", "Qikrana"));
        personList.add(new Person(21, "Naevys", "Qiren"));
        personList.add(new Person(23, "Filverel", "Nerina"));
        for(Person p : personList) { System.out.println(p); }

        // Trinomial task
        Trinomial t = new Trinomial(3,-4,5);
        System.out.println(t);
        System.out.println(t.Solution());
    }
}
```

CODE: Person.java

```
package Program;
public class Person {
    private int age;
    private String name;
    private String surname;

    private void setAge(int age) { this.age = age; }
    public int getAge() { return this.age; }

    private void setName(String name) { this.name = name; }
    public String getName() { return this.name; }

    private void setSurname(String surname) { this.surname = surname; }
    public String getSurname() { return this.surname; }

    Person(int age, String name, String surname)
    {
        this.setAge(age);
        this.setName(name);
        this.setSurname(surname);
    }

    @Override public String toString()
    {
        return String.format("=====\nName: %s\nSurname: %s\nAge: %d", this.getName(),
this.getSurname(), this.getAge());
    }

    private Person() {}
}
```

CODE: Trinomial.java

```
package Program;
import java.lang.Math;
public class Trinomial {
    private double a, b, c;
    private double delta;
    private int solutions;
    private double x1, x2;

    private void setA(double a) { this.a = a; }
    private void setB(double b) { this.b = b; }
    private void setC(double c) { this.c = c; }
    private void setDelta(double delta) { this.delta = delta; }
    private void setSolutions(int solutions) { this.solutions = solutions; }
    private void setX1(double x1) { this.x1 = x1; }
    private void setX2(double x2) { this.x2 = x2; }

    public double getA() { return this.a; }
    public double getB() { return this.b; }
    public double getC() { return this.c; }
    public double getDelta() { return this.delta; }
    public int getSolutions() { return this.solutions; }
    public double getX1() throws Exception { if(this.getSolutions() == 0) { throw new
Exception("This trinomial has no solutions"); } return this.x1; }
    public double getX2() throws Exception { if(this.getSolutions() == 0) { throw new
Exception("This trinomial has no solutions"); } return this.x2; }

    public String Solution()
    {
        String vessel = String.format("Number of solutions: %d", this.getSolutions());
        try
        {
            if(this.getSolutions() == 2)
            {
                vessel = vessel + String.format("\nx1 = %f\nx2 = %f", this.getX1(), this.getX2());
            }
            else if(this.getSolutions() == 1)
            {
                vessel = vessel + String.format("\nx = %f", this.getX1());
            }
            else
            {
                vessel = vessel + String.format("\nDelta = %f", this.getDelta());
            }
        }
        catch(Exception e) {} // Can't happen under any circumstances anyway
        return vessel;
    }
}
```

```

@Override public String toString()
{
    String vessel = String.format("%fx^2 ", this.getA());
    if(this.getB() > 0) vessel = vessel + String.format("+%fx ", this.getB());
    else vessel = vessel + String.format("%fx ", this.getB());
    if(this.getC() > 0) vessel = vessel + String.format("+%f ", this.getC());
    else vessel = vessel + String.format("%f ", this.getC());
    return vessel;
}

Trinomial(double a, double b, double c)
{
    this.setA(a);
    this.setB(b);
    this.setC(c);

    double d = Math.pow(b,2) - 4*a*c;
    this.setDelta(d);

    if(d > 0)
    {
        this.setSolutions(2);
        double rd = Math.sqrt(d);
        this.setX1((-b + rd) / (2*a));
        this.setX2((-b - rd) / (2*a));
    }
    else if(d == 0)
    {
        this.setSolutions(1);
        this.setX1((-b) / (2*a));
        this.setX2((-b) / (2*a));
    }
    else
    {
        this.setSolutions(0);
        this.setX1(0);
        this.setX2(0);
    }
}

private Trinomial() {}
}

```