Lab: Open/Closed and Liskov Substitution

Problems for exercises and homework for the "Java OOP Advanced" course @ SoftUni.

You are provided **Code Skeleton** for this Lab. **Import** skeleton project in **IntelliJ**, without changing its packages and file names. You can **add new classes** and **refactor** provided one. **Don't change names of packages and provided classes**

1. Extend ArrayList<T>

Import logic in **ExtendedArrayList** class. Like you see from its name it have to **extend ArrayList**, and like its parent it have to work with **generics**. There are **two** important **methods**, which you need to **implement**:

- `min()`
- `max()`

2. Stream Progress Info

Refactor code for this task, so **Stream Progress Info** can work with different kind of **Stream**. First make sure it work with **Music** too. Refactor code, so if in future **new kind of stream** need to be write, you will need **just to import one new class** with **getBytesSen**t() and **getLength()** methods in it.

3. Graphic Editor

Refactor code for this task, so **Graphic Editor can draw all kind of shapes** without checking, **what kind is concrete shape.** In the future new shapes will be added to system, so prepare the system for this moments. When you **add new shape**, you just have to **add new class and nothing more**.

4. Detail Printer

Refactor code for this task, so **Detail Printer** don't need to ask **what kind of employee is pass to it**. Detail Printer need just to print details for all kind of employees. When new kind of employee is added you will need just to **add new class and nothing more.**

5. Square

Like you know from Math, **square is a rectangle**. But in programming may be not the same. Look at skeleton for this task and find **where bug will be produced**. Refactor your code, so you **eliminate chance for bugs**. You have to make sure square is a rectangle after all, but be sure that **square is substitute for rectangle too.**