Lab 1: level 2 class activities

prof. dr. Irma Ravkic

This set of activities is for those who think they mastered the topics in Level 1 activity.

Lab 1.1 Generics trouble

Given the following classes:

```java
class Shape { /* ... */ }
class Circle extends Shape { /* ... */ }
class Rectangle extends Shape { /* ... */ }

class Node<T> { /* ... */ }
```

will this code compile? If not, why not?

```java
Node<Circle> nc = new Node<>();
Node<Shape>  ns = nc;
```

Try to write this. Create a new project. Create these classes (you don't need to add any methods to it). Try to check whether the code will compile. Why is this not working given that we can write the following?

```java
Shape nc = new Circle();
```

Please read this in order to learn why the code above is not working:
https://docs.oracle.com/javase/tutorial/java/generics/inheritance.html

Lab 1.2 Generics + Interfaces

Define an interface Appendable that includes an append method. Then define two classes, *MyString* and *MyList*, which both implement Appendable. However, there are constraints: a MyString can be appended to a *MyString*, and a *MyList* to be appended to a *MyList*, but not a *MyString* to a *MyList*, or a *MyList* to a *MyString*. Here is one definition of Appendable:

```java
interface Appendable {
    Appendable append(Appendable a);
}
```

What is wrong with this definition given this task? What is a correct one? Also write a definition for a class MyString that uses the revised definition of Appendable. Test your solution with the following piece of code:

```java
public static void main(String[] args) {
       MyString s = new MyString("A");
       Appendable sNew = s.append(new MyString("B"));
       System.out.println(sNew);
   }
```