Homework 0

Elliott Pryor

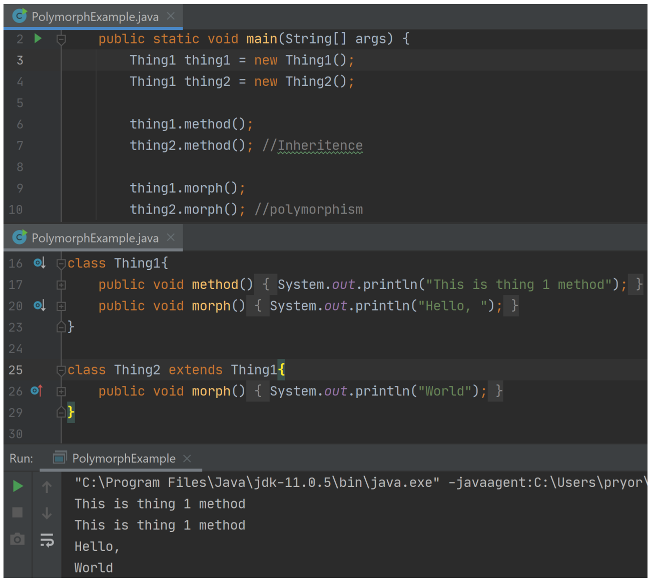
21 Aug 2020

Problem 1

1. What is polymorphism? Provide an example using source code
2. What is inheritance? Describe the differences between public vs private inheritance
3. What is encapsulation?
4. What is the difference between static vs dynamic binding?

---

1. Polymorphism is changing the function of a method in a parent class via inheritance. Figure 1 shows a very simple example of polymorphism.



```java
    public static void main(String[] args) {
        Thing1 thing1 = new Thing1();
        Thing1 thing2 = new Thing2();

        thing1.method();
        thing2.method(); //Inheritence

        thing1.morph();
        thing2.morph(); //polymorphism
```

```java
class Thing1{
    public void method() { System.out.println("This is thing 1 method"); }
    public void morph() { System.out.println("Hello, "); }
}

class Thing2 extends Thing1{
    public void morph() { System.out.println("World"); }
}
```

```
"C:\Program Files\Java\jdk-11.0.5\bin\java.exe" -javaagent:C:\Users\pryor\
This is thing 1 method
This is thing 1 method
Hello,
World
```

Figure 1: Example of polymorphism

2. Inheritance is when one class extends another and inherits the methods and variables from the parent class. The child class is a more specific type of the parent class. In public inheritance public methods in the parent class remain public methods in child class, but in private inheritance public methods in the parent class become private methods in the child class. In figure 1 Thing1 is the parent class and Thing2 is the child class.

3. Encapsulation is the restriction of access to a method or variable within a class. This limits unwanted access and can help protect functionality. A common example is the use of getters and setters for encapsulated variables.

4. Static binding is fixed and can't be changed (static). This would be done at compile-time, and cannot be overriden. Dynamic binding is loose and done during run-time. Dynamically bound methods can be changed during runtime (inheritance) [3]

Problem 2

i. Provide a definition for a prescribed lifecycle, and name two types of prescribed lifecycles

ii. For each type provide an example (i.e. a use case)

i. A prescribed lifecycle is a processes of development. It outlines a methodology for how a piece of software can go from an idea to deployment. One example is the waterfal model, another example is the evolutionary model.

ii. The waterfall model is used for slow development projects. For example an operating system, we do not want a lot of changes, but instead go from design to deployment in one single chain. The evolutionary model is a cycle. It goes from design to deployment back to design very fast, so is good for prototyping.

Problem 3

What does low coupling and high cohesion mean?

Coupling is the amount of interdependence between modules. Low coupling is preferred because it means that modules can be changed without much influence to the rest of the program.

Cohesion is how alike processes are within a module. If they all serve a common goal this leads to high cohesion as they are very alike and united. If a module has a more un-related subroutine this leads to low cohesion. [2, 1]

References

[1] software engineering — coupling and cohesion - geeksforgeeks, January 2020. URL: `https://www.geeksforgeeks.org/software-engineering-coupling-and-cohesion`.

[2] Vijay Menon. What are some examples of coupling and cohesion for class and object?, 11 2017. URL: `https://www.researchgate.net/post/What_are_some_examples_of_coupling_and_cohesion_for_class_and_object`.

[3] Chaitanya Singh. Static and dynamic binding in java, May 2013. URL: `https://beginnersbook.com/2013/04/java-static-dynamic-binding`.