Roberto Andrade

CST-239-0500

02/13/2024

Video: <https://youtu.be/vWT6BRh8ni8>

Topic 2-1:

In this screenshot It shows how we have created the two types of superheros and how we made them attack each other. We used a random object to create the damage and health, and used the attack to cause damage to the health till one is dead. In the Console you can see that the superhero’s are causing damage to each other till one of them defeats the other. Different methods were used to pass through the damage and to see which hero is dead first to end the game.

A screenshot of a computer

Description automatically generated

Topic 2-2

In this next screenshot, the console shows how we printed out that we were in the method fireWeapon() inside the class Weapon. We also sent the argument of 10 and 5 for both different types of weapons. That’s how we show that we are in a certain method, that got sent a specific argument for popwer.

A screenshot of a computer

Description automatically generated

Topic 2-2

In this next screenshot we show how to override the method fireWeapon() inside the class for weapon. We can see that the class is an extended class of the super class weapon, and when being called, it will override the output, and instead print what’s in Gun.fireWeapon() and for Bomb.fireWeapon(). This is inheritance but sometimes we need to modify the implementation of a method.

A screenshot of a computer

Description automatically generated

Topic 2-2

In this screenshot we see how overload works and what the Console will show when it is printed out. Overloading happens when there are two or more versions of the same method in the class. Here we see that there is the two methods for fireWeapon() but one has a parameter while the other one doesn’t. So it will print out the first methods that takes in a parameter, then the one without the parameter will also be called, and then print out the base class method using the super keyword.A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

Topic 2-2

The compiler error shows that Bomb and Gun classes cannot access the methods inside the Weapon class since the keyword Final makes them non-changeable.

A screenshot of a computer

Description automatically generated

In the next screenshot, I am thinking that the compiler error is meaning that the methods fireWeapon() in the classes Gun and Bomb cannot override the method in Weapon because of the Final keyword in the fireWeapon() method.

A screenshot of a computer

Description automatically generated

In the next screenshot, I think that the complier error is due to the keyword abstract being in the method fireWeapon(), and since it is in the superclass Weapon, it cannot be used since abstract classes can’t be instantiated but can be in subclasses.

A screenshot of a computer

Description automatically generatedsubclassed.

Topic 2-3

In this screenshot, we see how we complied the code and printed out answers into the console. In the console we see, how we used the if-statements but the expression we used equals() which is in a way “==” but we choose what exactly it should be equal to. Also, we us the to.String() method to print out the objects, and you can see the hashcode of each object, and see the difference between the name objects.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Topic 2-3

1)

Hello I see that the toString() has been changed and that it overrides test objects, to send out the actual names of the objects and not the hashcode. I am a little confused though on where did the getClass() come from.

2)

I have used the equals() in past programs that I have created, and I think it was in the first activity, but I think I can use the method to show which item is being wanted in the shopping cart and what quantity, and take that away from the inventory. Also the toString() I would like to find a way to override when created the salable, I just have been having trouble with them. I think I can use them in different ways, but I think the interaction of the user from the StoreFront to the Invenrtory Manager will be very helpful.

3)

Overriding is useful in how we want to implement the input of the user and output what is the best thing for the users input. If we want to use the same method that is in the superclass, we can override that method in the subclass and change the method. I think inheritance is such a great way to save time when compiling code and also keeps the class well organized for when we have to overriding.

Topic 2-4:

Breakpoints:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Inspecting Variables:

A screenshot of a computer

Description automatically generated

Stepping Task:

A screenshot of a computer

Description automatically generated

Call Stack

A screenshot of a computer

Description automatically generated