Inspiring Greatness

ISD

INDEPENDENCE SCHOOL DISTRICT

Computer Science Virtual Learning

HS Computer Science A

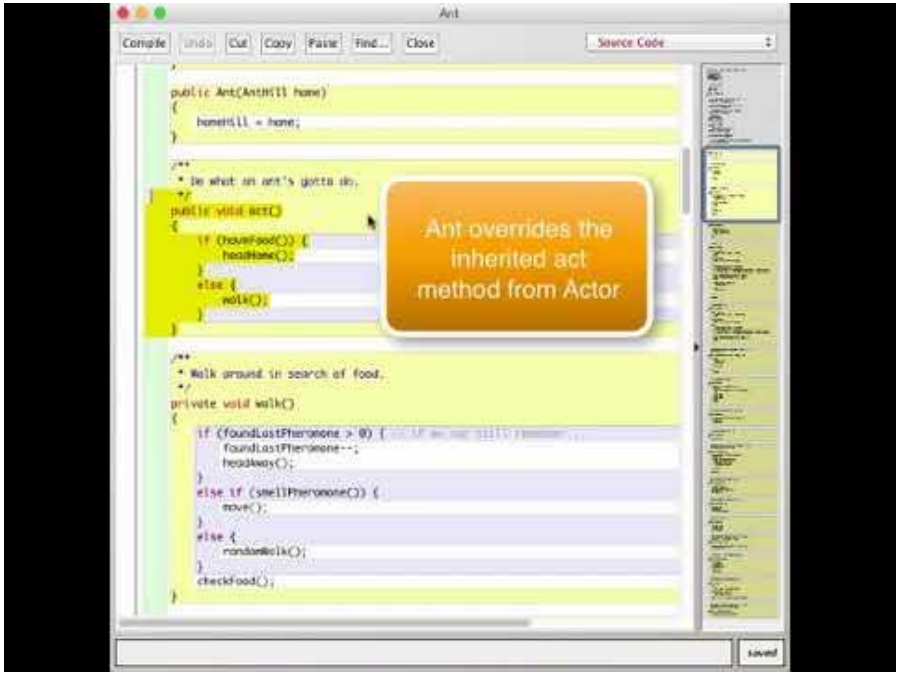
May 18th, 2020

Lesson: What is Object-Oriented Programming?

Objective/Learning Target:

Understanding what Object-Oriented Programming is and how to apply it using Java

What is an Object-Oriented Programming?



What is Object-Oriented Programming?

Object-oriented programming has three main features: objects, inheritance, and polymorphism.

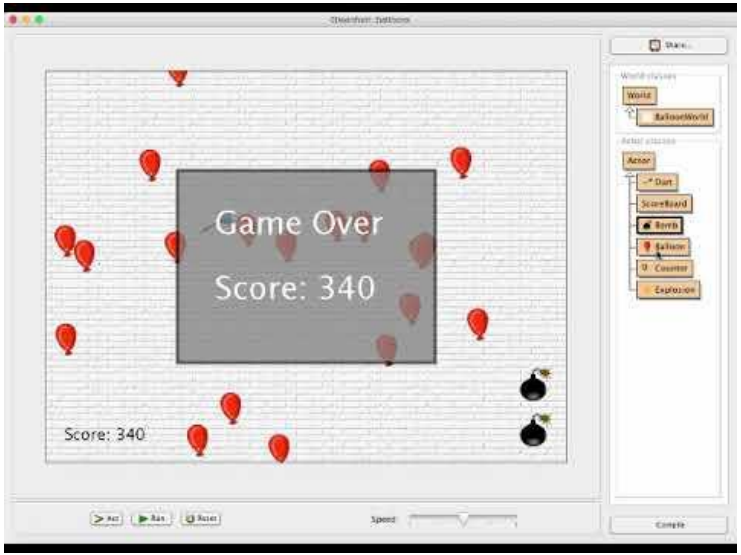
Objects have data (fields) and behavior (methods) and do the work in an object-oriented program. Objects are created by classes. A class defines the data (fields) and behavior (methods) for all objects of that class. You can create many objects from the same class.

What is Inheritance?

Inheritance allows for cleaner code since a class can inherit fields and behavior from another class instead of copying code from class to class. The parent class is specified using the extends keyword in the class declaration. The class that is extending the parent class is called the child class. In the ants scenario the Ant class inherits from the Actor class. The Ant class is the child class and the Actor class is the parent class. The Ant class inherits the act method from the Actor class, but overrides it by creating a method with the same signature that will be executed instead of the parent's method. This allows the Ant class to modify what an Ant object does when it acts.

What is Polymorphism?

Polymorphism allows for specialized behavior based on the run-time type. It also removes the need for conditional execution based on the type. Java uses inheritance-based polymorphism where a parent class has a method that the children classes override to provide specialized behavior. In the Ant scenario the Balloon and Bomb classes inherit from the Actor class and both override the act method. The world contains a list of all Actor objects in the world and tells each to act. What happens when an Actor object acts depends on the class that created it (the run-time type).



Objects and Classes

In object-oriented programming the programmer writes a class which defines what all objects of the class know and can do. You can think of the class as like a cookie cutter or factory that produces objects. All objects created by the same class have the same fields and methods. A field is something the object knows about itself and a method is a thing the object can do.

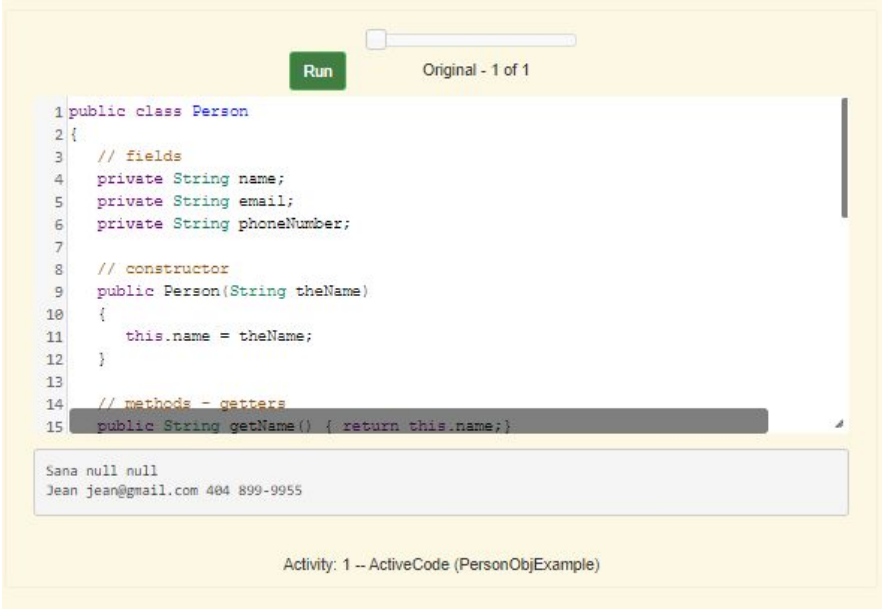
A class also has constructors which initialize the fields when the object is created. A class can also have a main method which can be used to test the class.

Check Your Understanding: Person Class

What should we want to know about a person? What we want to know depends on what problem we are trying to solve. In one situation we might want to know the person's name and phone number and email. We need ways to get and set the data (fields) so we create getters and setters.

Go to:
https://runestone.academy/runestone/books/published/apcsareview/OOBasics/objectsAndClasses.html

Scroll to the Person Class section and Modify the code above to add more constructors. Also modify the main method to test the new constructors.

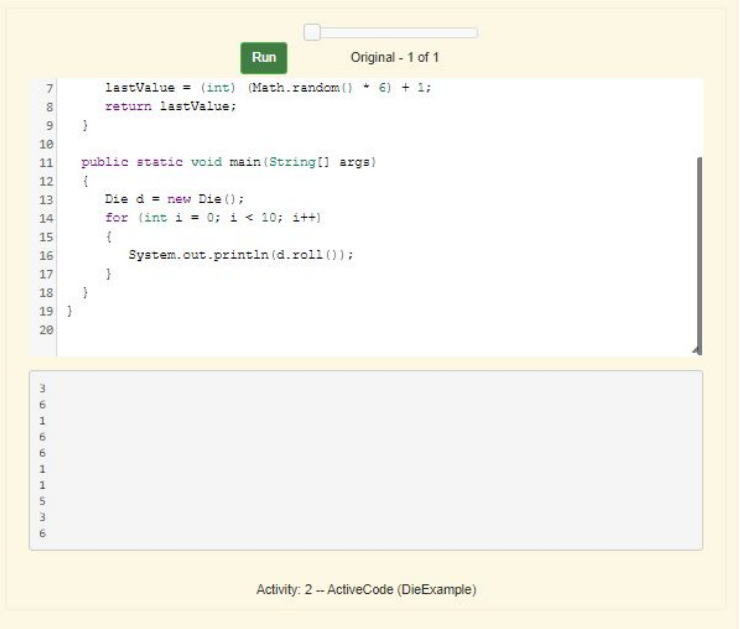


Check Your Understanding: Die Class

What if you wanted to represent a die which has 6 sides and you can roll it? You might also want to keep track of the last value rolled. Does the following class have everything it needs?

Go to:
https://runestone.academy/runestone/books/published/apcsareview/OOBasics/objectsAndClasses.html

Scroll to the Die Class section. Can you modify the Die class to keep a record of all the values this dice has rolled? How would you do that?

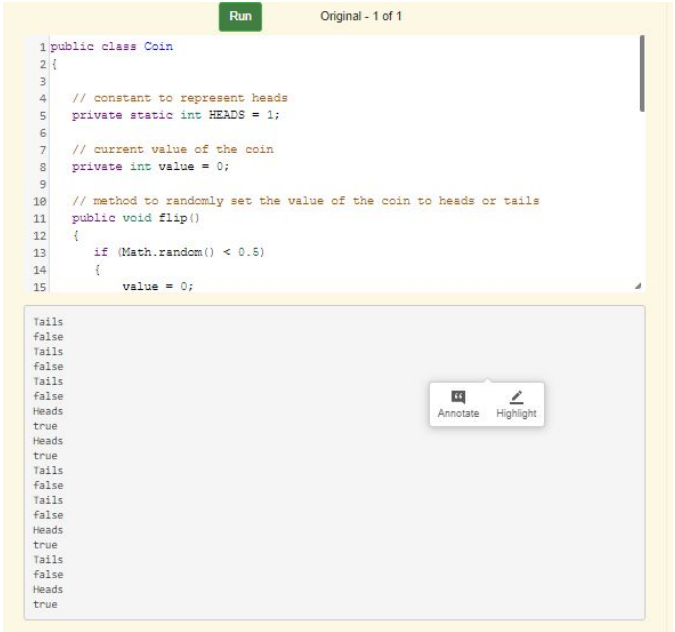


Check Your Understanding: Coin Class

What if you just wanted to simulate flipping a coin? What would you need the objects of the class to know and do? You would want to flip the coin (set it randomly to heads or tails) and ask if the value is heads or tails. See the class below for one way to do this. Notice the use of a constant for HEADS. Any field that is declared to be static can't be changed and so is a constant.

Go to:
https://runestone.academy/runestone/books/published/apcsareview/OOBasics/objectsAndClasses.html

Scroll to the Coin Class section. Modify the class to add a isTails method that returns true when the value is not heads.



For More Resources and to Check Answers

Go to: https://runestone.academy/runestone/books/published/apcsareview/OOBasics/objectsAndClasses.html