Greenfoot: Food Chains Cheat Sheet

Greenfoot Help

Method	Purpose	Example	Explanation

`super(int x, int y, int z);`	This is a java method which allows access to methods from the specific objects' superclass.	`super(600, 400, 1);`
		`written in a World object i.e. MyFirstWorld.`
			This sets a world with a grid of 600 by 400 cells, where each cell contains 1 by 1 pixels.

`"NameOfClass" (name of Object) = new "NameOfClass"();`	Creates a new object within the given class.	`MainCharacter frog = new MainCharacter();`	This creates a new object called frog in the class MainCharacter.

`addObject(Actor object, int x, int y);`	This allows you to place an object in the specific World.	`addObject(frog, 1, 1);`
		`written in a World Object i.e. MyFirstWorld.`
			This places a previously created object named frog in cell (1,1).

`setRotation(int rotation);`	Sets the rotation of an object.	`setRotation(90);`
		`written in an Actor object.`
			This sets the rotation of an object to 90 degrees i.e. facing downwards. 0 = right, 90 = down, 180 = left, 270 = up.

Greenfoot: Food Chains Cheat Sheet

Greenfoot Help

Method	Purpose	Example	Explanation

`move(int distance);`	This makes an object move a given distance (in cell size) in the direction it is facing.	`move(1);`
		`written in an Actor object.`
			This makes an Actor object move a distance of 1 cell in the direction it is facing.

`isTouching(Class cls);`	Checks whether this actor is touching any other objects of the given class.	`isTouching(Collectables.class);`
		`written in MainCharacter object.`
			Checks if the MainCharacter is touching an object of the class Collectable.

`removeTouching(Class cls);`	Removes anything in a given class which is touching the object this method is written in.	`removeTouching(Collectables.class);`
		`written in MainCharacter object.`
			Removes from the world any objects in the Collectables Class that the MainCharacter touches.

`Greenfoot.getRandomNumber(int limit)`	Return a random number between 0 (inclusive) and limit (exclusive).	`Greenfoot.getRandomNumber(4);`	Returns a random number between 0 and 3. Does not include the number entered i.e. 4!

Greenfoot: Food Chains Cheat Sheet

Greenfoot Help

Class
Inheritance
Method
Object
Compile
Documentation

Class - A Class is like an object constructor, or a "blueprint" for creating objects.

Object - An object is an instance of a class.

Inheritance - Objects are often very similar. They share common logic. But they're not entirely the same. Inheritance enables new objects to take on the properties of existing objects. A class that is used as the basis for inheritance is called a superclass, base class or parent class. A class that inherits from a superclass is called a subclass, derived class or child class.

Compile - convert (a program) into a machine-code or lower-level form in which the program can be executed.

Method - A method is like an instruction that can be called on the class or object.

Documentation - information that describes the product to its users. It consists of the product technical manuals and online information.