CODEGYM

Java developer in 12 months

Mentor-supported training program from CodeGym

MODUEL 2. JAVA CORE

Lesson #8

Peculiarities of calling constructors.

Static block.

CODEGYM

Lesson plan

- Order of calling constructors
- Class loading order
- Variable initialization order

CODEGYM

Order of calling constructors

Process of creating an object:

1. First, memory is allocated for all variables – class fields.
2. Then comes the initialization of the base class.
3. Then all variables are assigned values, if they are specified.
4. Finally, the constructor is called.

If a class does not have any constructors, it will be created automatically.

Class loading order

The Java machine does not load classes without a need.

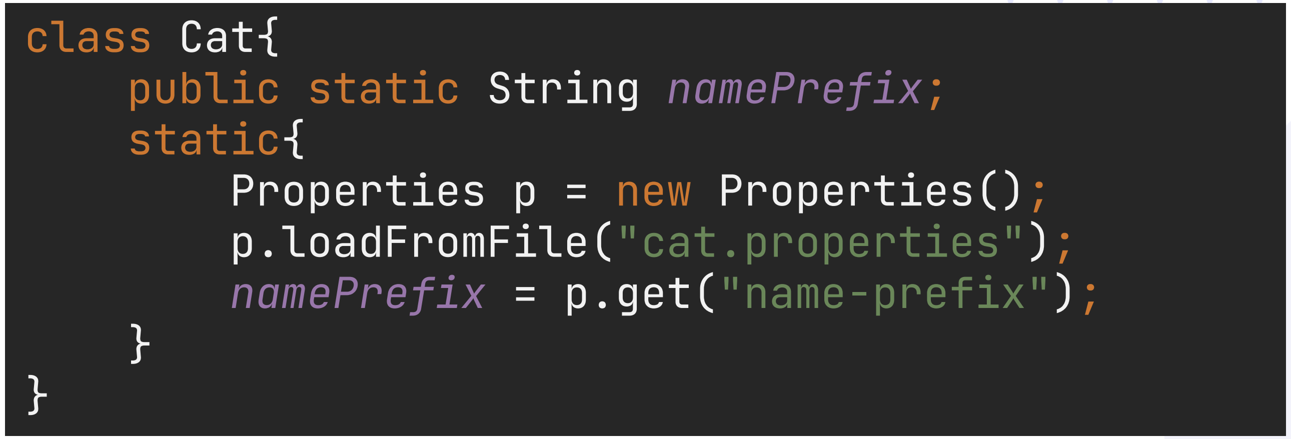
As soon as a class is called somewhere in the code, the Java machine checks to see if it is loaded. If not, it loads and initializes it.

Initializing a class means assigning values to all its static variables and calling of all static blocks.

Static block

If complex code needs to be executed to initialize an object's variables, we can do it in the constructor.

Static variables do not have this capability. But since the need for this remains, you can add a static block or blocks to classes, which are essentially equivalent to calling static constructors.



```
class Cat{
    public static String namePrefix;
    static{
        Properties p = new Properties();
        p.loadFromFile("cat.properties");
        namePrefix = p.get("name-prefix");
    }
}
```

MODULE 2. JAVA CORE

Initialization order of variables

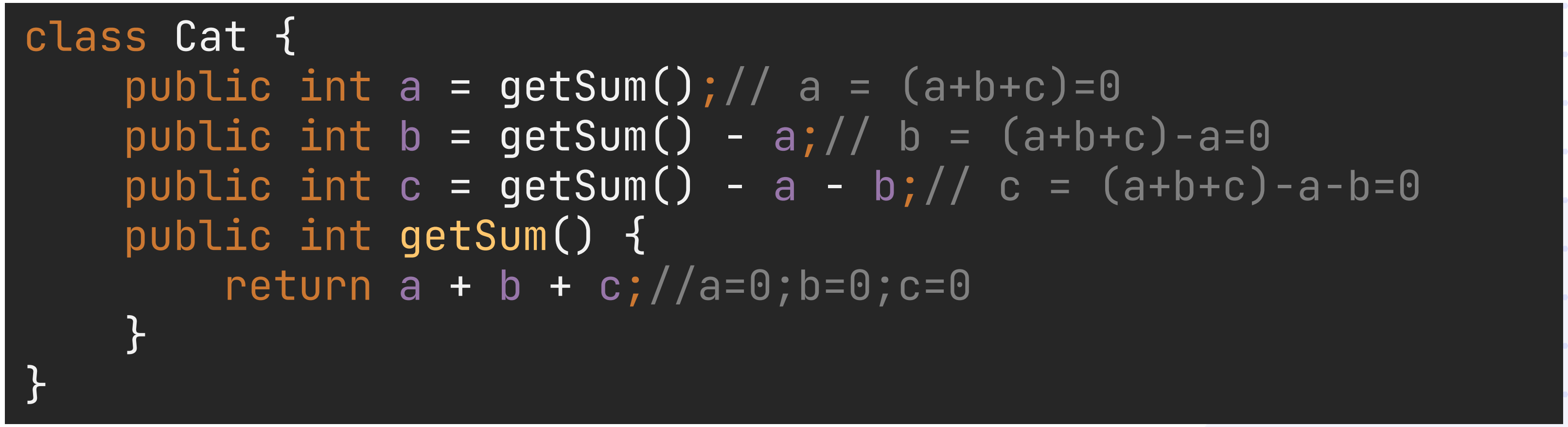
The order of declaration between methods and fields of a class is unimportant.

The class is loaded from top to bottom, so it's important that a class field only refers to other fields that are already loaded.

In the example below, field b can refer to a, but knows nothing about c.

```
class Cat {
    public int a = 5;
    public int b = a + 1;
    public int c = a * b;
}
```

When variables are created, they are set default values.



```
class Cat {
    public int a = getSum();// a = (a+b+c)=0
    public int b = getSum() - a;// b = (a+b+c)-a=0
    public int c = getSum() - a - b;// c = (a+b+c)-a-b=0
    public int getSum() {
        return a + b + c;//a=0;b=0;c=0
    }
}
```

MODULE 2. JAVA CORE

CODEGYM

Homework

Module 2

Level 8

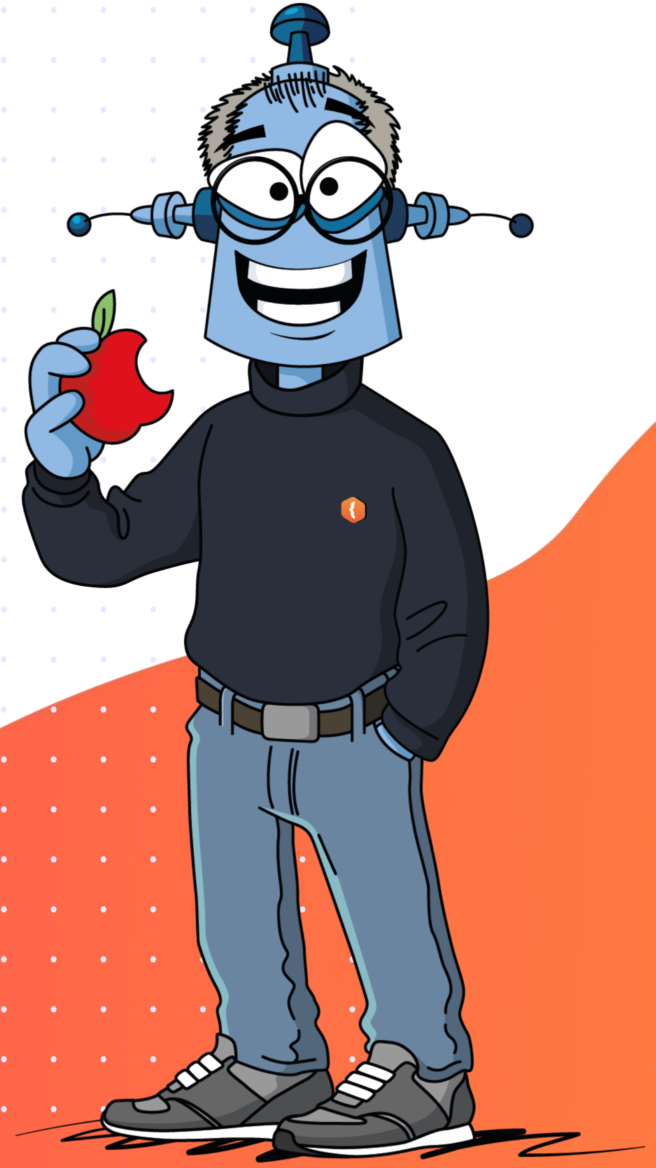
Peculiarities of calling constructors.

Static block.

MODULE 2. JAVA CORE

CODEGYM

Questions and Answers



MODULE 2. JAVA CORE