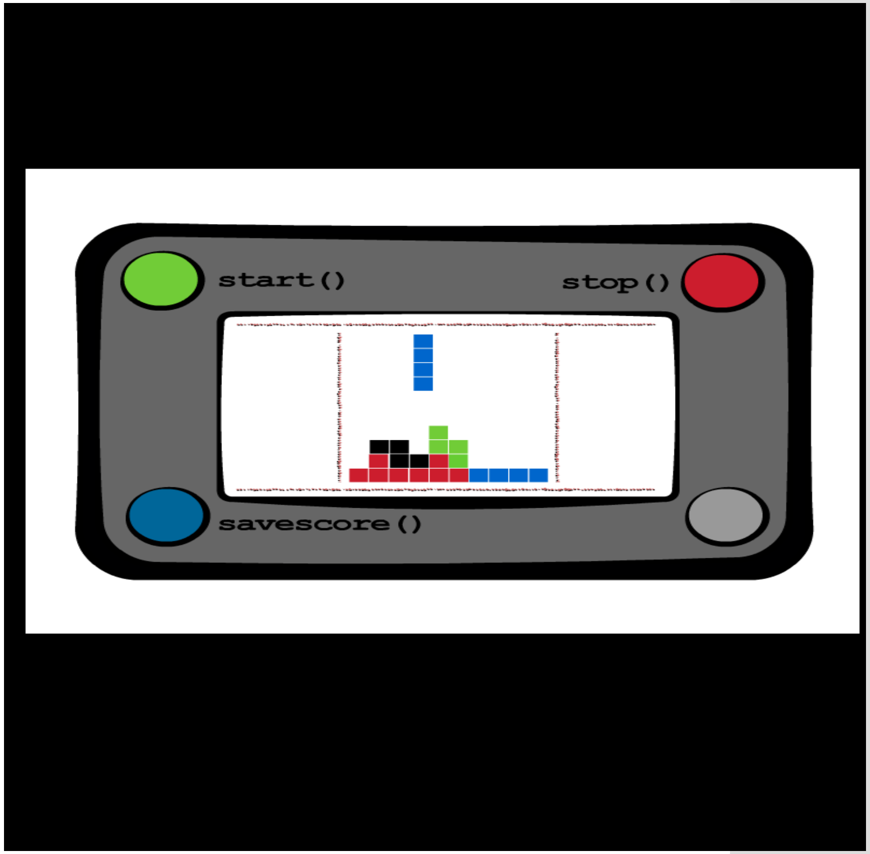
Chapter 3: Meet Classes

Java Programming for Kids
http://yfain.github.io/Java4Kids/
by Yakov Fain was used to prepare this course

Lesson 1: Classes and Objects

The Class VideoGame

```
class VideoGame {

int price;

void start () {
// Code to start the game
}

void stop () {
// Code to stop the game
}

}
```

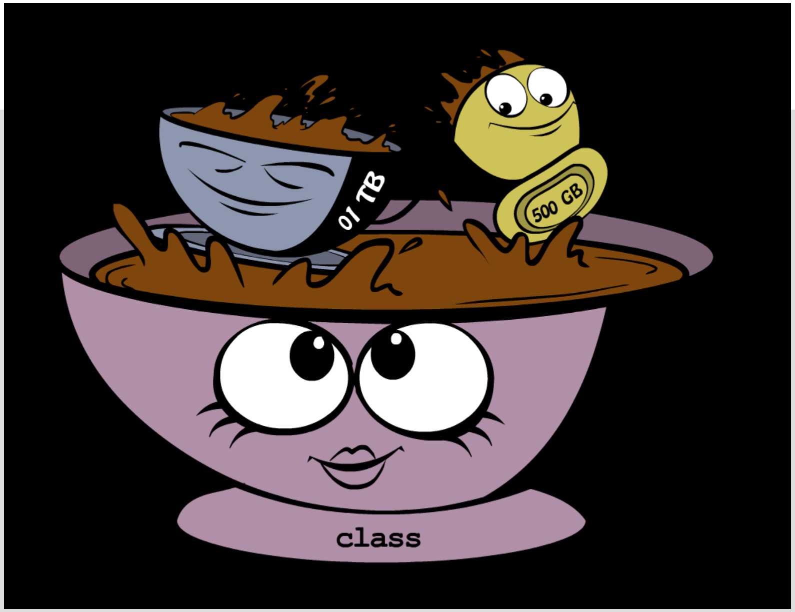


The Class PlayStation4

```
class PlayStation4{

int hardDiskSize;
// Some other attributes and methods

void shareOnFacebook(){
// Code to share on Facebook
}

void shareOnTwitter(){
// Code to share on Twitter
}
}
```

Creating PlayStation Objects

```
public class CreatePlayStation4Objects {

   public static void main(String[] args) {
      PlayStation4 firstPlayStation =
                                 new PlayStation4();
      firstPlayStation.shareOnFacebook();
   }
}
```



Java Data Types and Variables

We need to declare:

- ***the variable type***
- ***the variable name***

```
➔String theNameOfMyFriend;
➔int hardDiskSize;
```

Java Data Types and Variables

The method parameters (a.k.a. arguments) are also stored in variables.

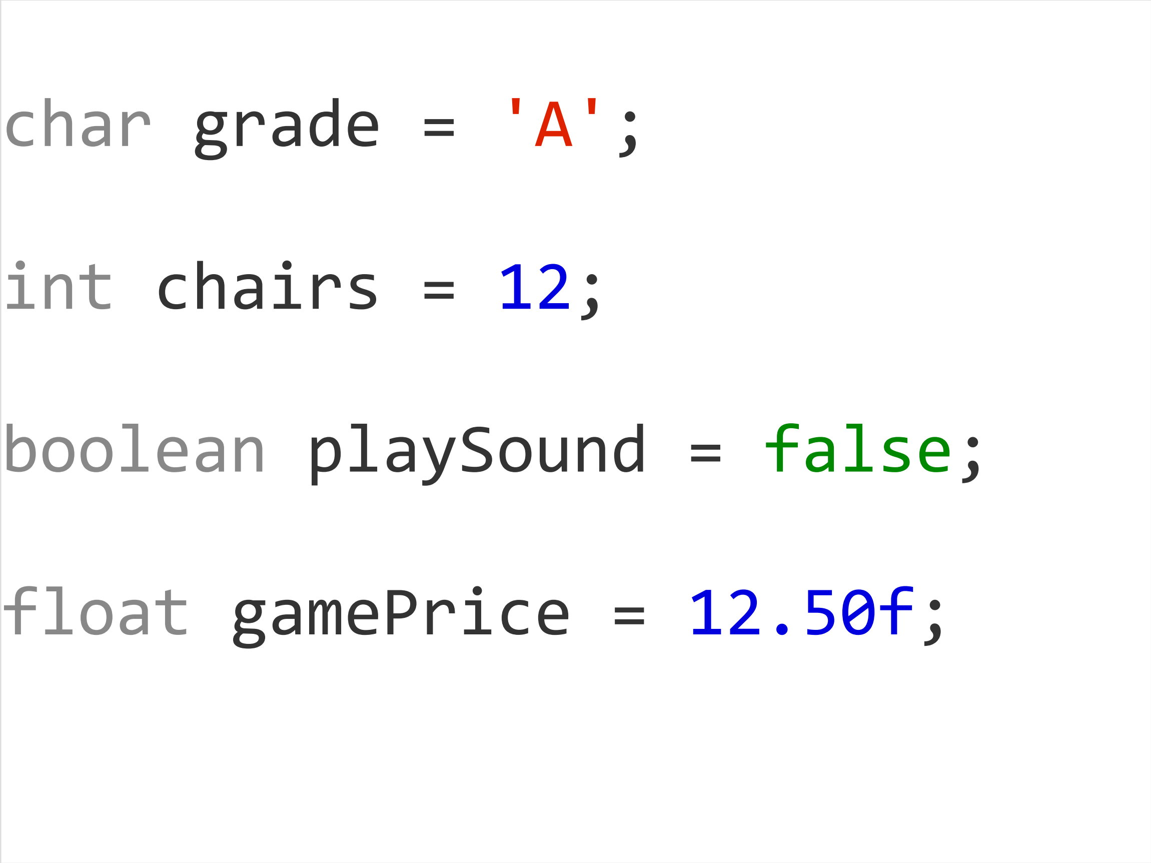
- `void saveScore(String playerName, int score);`

Java Data Types and Variables

Simple Java data types are called primitives.

- `int myScore=5;`
- `myScore=myScore+1;`

Variable Initialization



```
char grade = 'A';

int chairs = 12;

boolean playSound = false;

float gamePrice = 12.50f;
```

Classes as Data Types

- VideoGame myFirstGame;
- Pet myPet;
- Integer numberOfPlayers;
- Double mySalary;
- Boolean mailAlert;
- Etc.

Quiz

We’ve used a variable of type int to store the price in the class VideoGame.

Is there a more suitable primitive data type for storing prices?

Objects

```
public class CreatePlayStation4Objects {

public static void main(String[] args) {

// create one instance of PlayStation4 class
PlayStation4 firstPlayStation = new PlayStation4();

// call the method shareOnFacebook
firstPlayStation.shareOnFacebook();
}

}
```

Creation of a Pet

```java
public class Pet {

int age;
float weight;
String color;

public String talk(String aWord){
String petResponse = "OK!! OK!! " +aWord;
return petResponse;
}

}
```

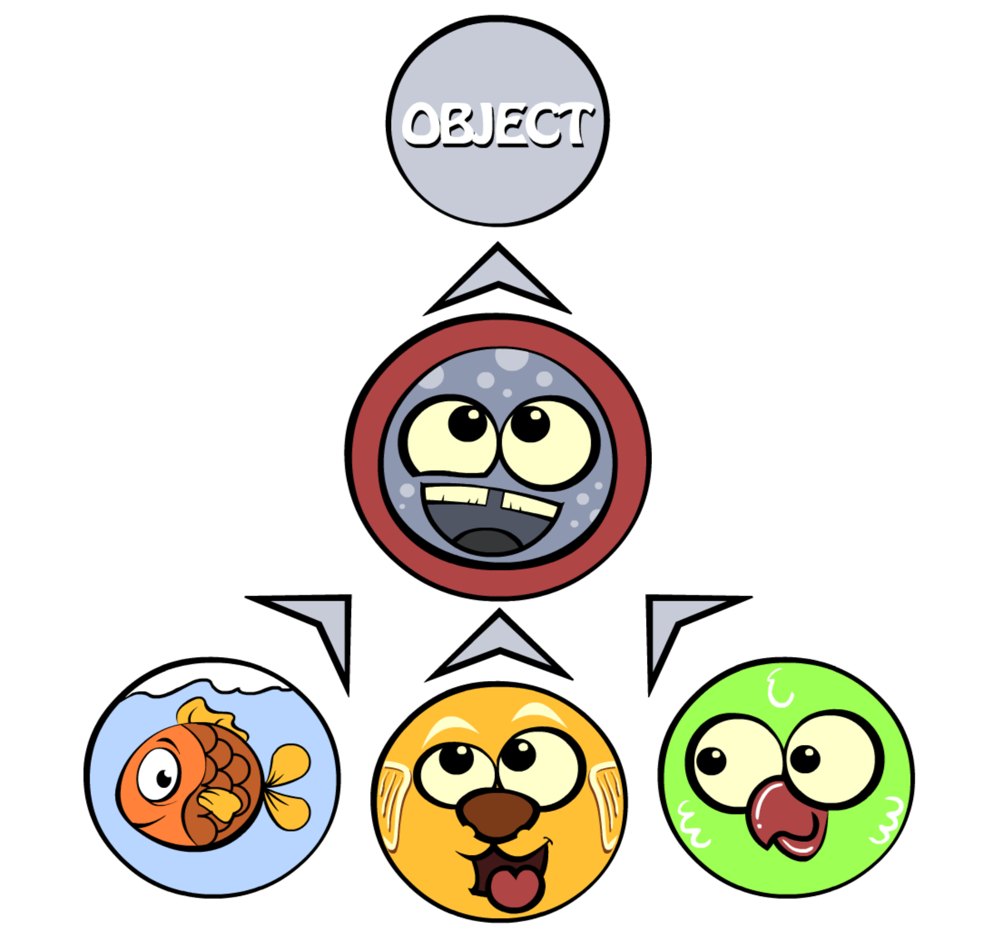
Challenge Yourself!

- Create a class ***Dog***
- Declare an attribute ***name*** in the class Dog
- Declare a method ***bark*** in the class Dog
- Create an instance of the class Dog
- Call the method bark

Lesson 2: Inheritance

Fish Don't Talk

```
public String
talk(String something){
return "Don't you know
that fish do not
talk?";
}

class Fish extends Pet{
}
```

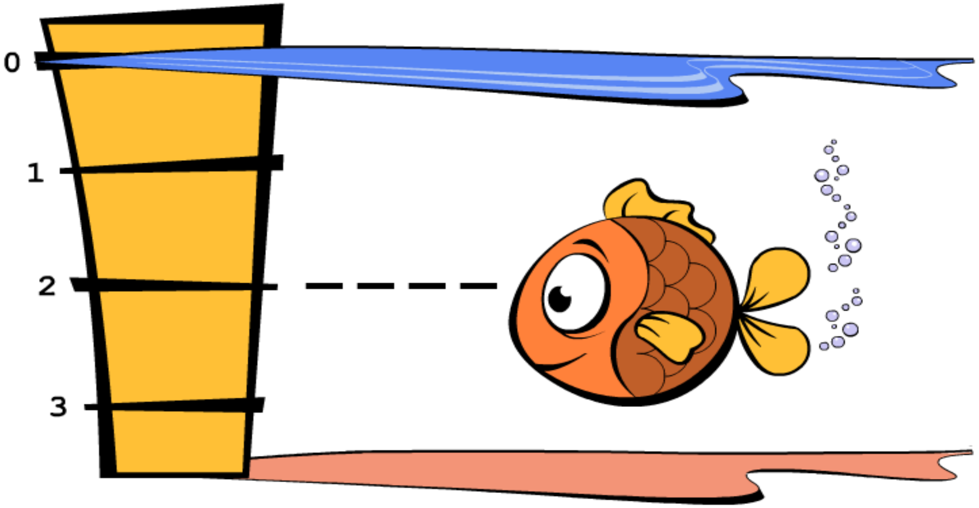


Not all pets can dive

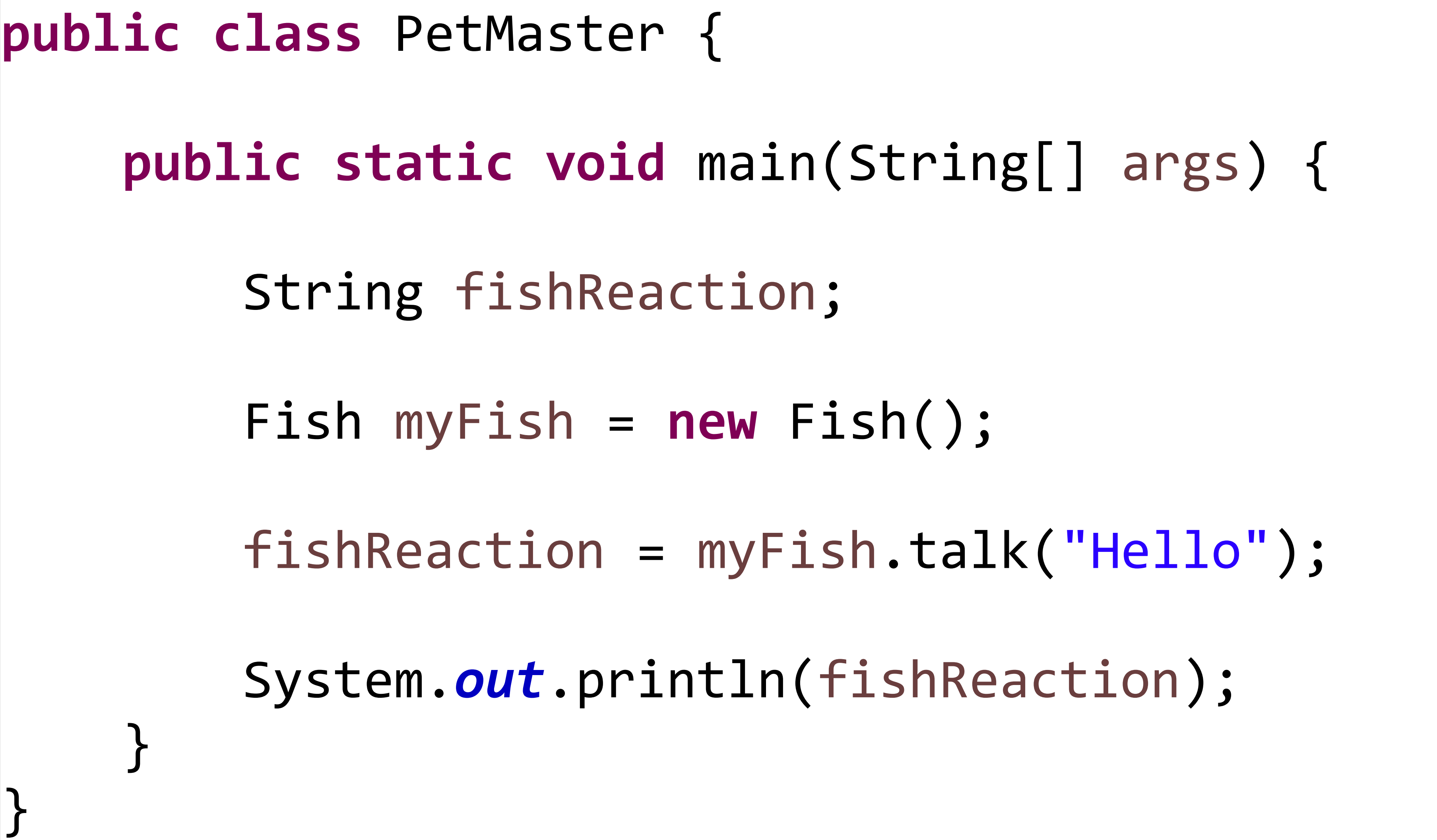
```
public class Fish extends Pet {

int currentDepth=0;

public int dive(int howDeep){

currentDepth=currentDepth + howDeep;

return currentDepth;

}
}
```

Fish Master

```
public class FishMaster {
public static void main(String[] args) {
Fish myFish = new Fish();

myFish.dive(2);
myFish.dive(3);
myFish.sleep();

}

}
```



Method Overriding



```
public class PetMaster {

    public static void main(String[] args) {

        String fishReaction;

        Fish myFish = new Fish();

        fishReaction = myFish.talk("Hello");

        System.out.println(fishReaction);
    }
}
```

Challenge Yourself!

1. Create a Java class Car with the following methods:

```
- public void start()
- public void stop()
- public int drive(int howlong)
```

2. Write another class CarOwner with the method main that creates an instance of the object Car and call its methods. The result of each method call has to be printed using System.out.println();

Challenge Yourself!

3. Create a subclass of Car named JamesBondCar and override the method drive there. Now use the following formula to calculate the distance: distance = howlong*180;

4. Create an instance of the class JamesBondCar in the main method of CarOwner. Re-run the program CarOwner to see that it calls the overridden method.

Extra Challenge

- Use FreeTTS, a Java Speech API implementation, to make your pets talk with a human-like voice
- http://www.adam-bien.com/roller/abien/entry/merry_christmas1