SCC.111 Software Development – Lecture 37: Case Study - GameArena

Adrian Friday, Hansi Hettiarachchi and Nigel Davies

Introduction

Lancaster University

- In the last lectures, we:
  - Introduced 3 core object-oriented programming concepts:
  - **Inheritance, Polymorphism, and Interfaces**
  - We also discussed abstract classes and methods, method overriding and method overloading.

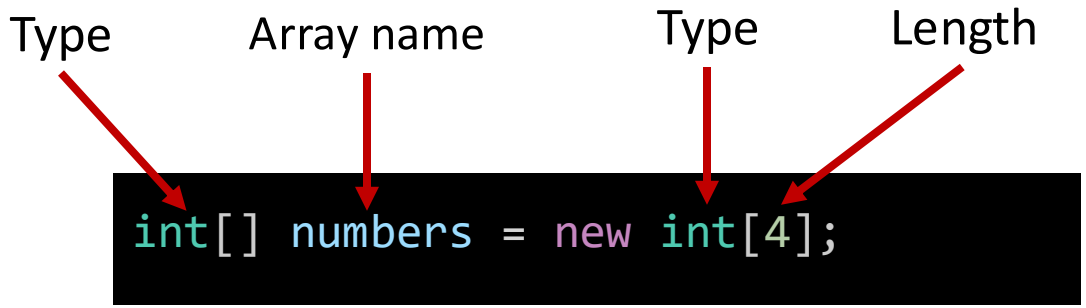
- Today we will
  - **Revise** all these concepts in connected examples
  - **Practice** these concepts with a case study on GameArena API

SCC111 Assessment Task

Lancaster University

Any questions?

Arrays syntax in Java



0	0	0	0

```
String[] names = new String[4];
```

null	null	null	null

```
Car[] cars = new Car[4];
```

Arrays syntax in Java

Lancaster University

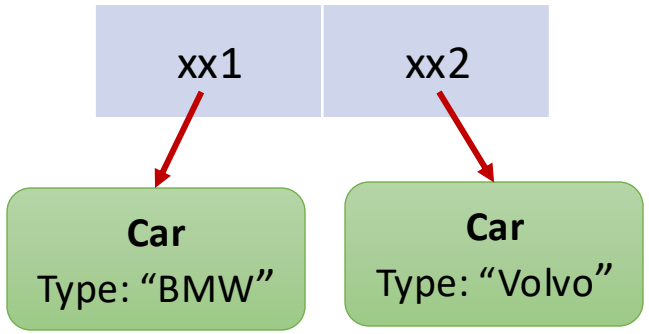
```
int[] numbers = new int[4];

numbers[1] = 5;
```

0	5	0	0

```
int[] numbers = {1, 5, 10, 20};
```

1	5	10	20

```
Car[] cars = {new Car("BMW"), new Car("Volvo")};
```



Loop through Arrays

Lancaster University

```
Car[] cars = {new Car("BMW"), new Car("Volvo")};
```

- For Loop:

```
for(int i=0; i<cars.length; i++){
    cars[i].printMilege();;
}
```

- For-Each:

```
for(Car car : cars){
    car.printMilege();
}
```

- ✓ Does not require a counter
- ✓ More readable
- ✓ Good for reading values

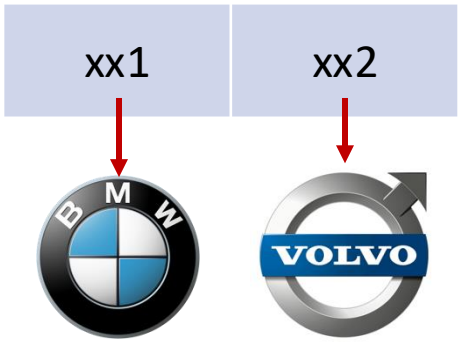
Loop through Arrays

Lancaster University

```
Car[] cars = {new Car("BMW"), new Car("Volvo")};
```

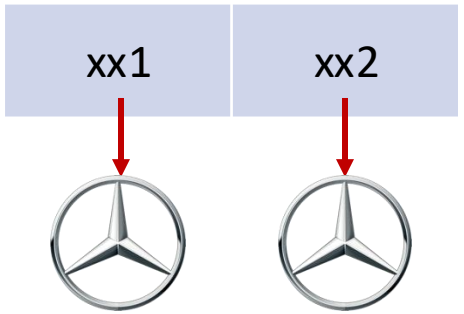
```
for(Car car : cars){
    car = new Car("Mercedes");
}
```

(A)



OR

(B)



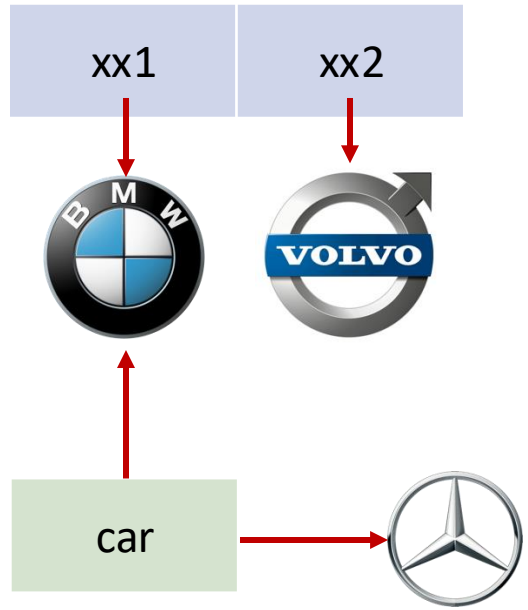
Loop through Arrays

Lancaster University

```
Car[] cars = {new Car("BMW"), new Car("Volvo")};
```

```
for(Car car : cars){
    car = new Car("Mercedes");
}
```

- Iteration 1:

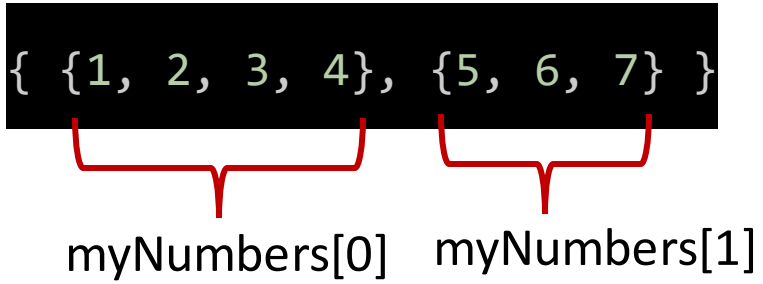


Multi-dimensional arrays

Lancaster University

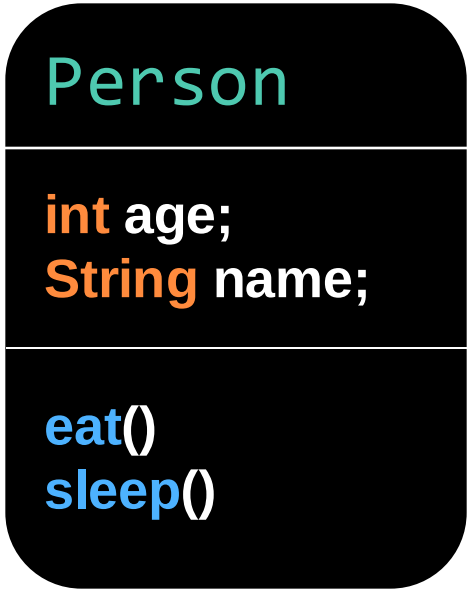
```
int[][] myNumbers = { {1, 2, 3, 4}, {5, 6, 7} };
```


```
for (int i = 0; i < myNumbers.length; i++) {
    for (int j = 0; j < myNumbers[i].length; j++) {
        System.out.print(myNumbers[i][j] + " ");
    }
    System.out.println();
}
```



Classes and Methods

Lancaster University



- Encapsulation:
  - **private –** *even* to children**, public** - *anyone***,** and **protected** – *class and children only**
- Constructor : method that create an object instance

```
public Person(){}
```

Overloading!

```
public Person(String name, int age){
    this.name = name;
    this.age = age;
}
```

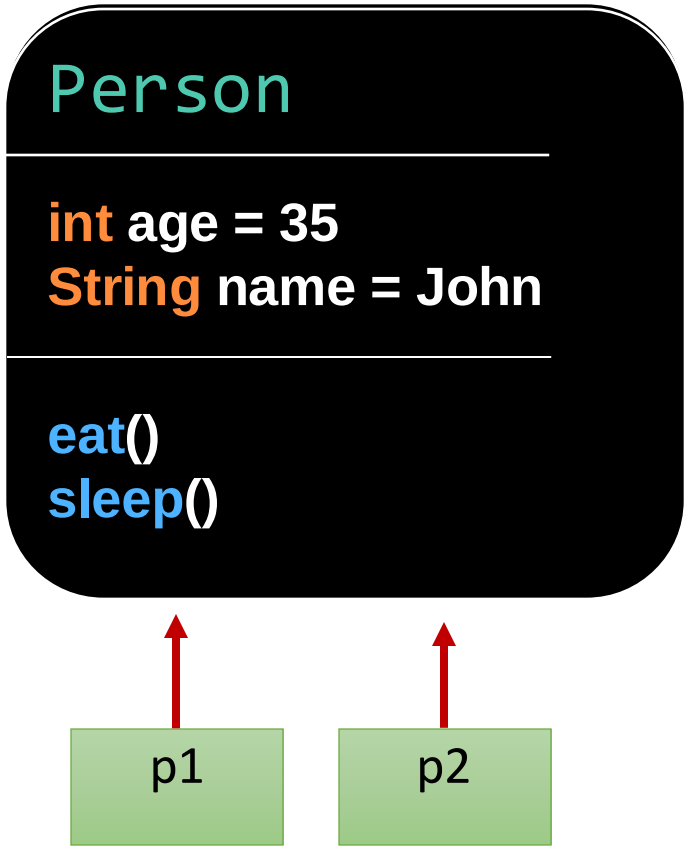
```
Person p = new Person();
```

```
Person p = new Person("John", 35);
```

*** We will cover it in the demo**

Object References

Lancaster University



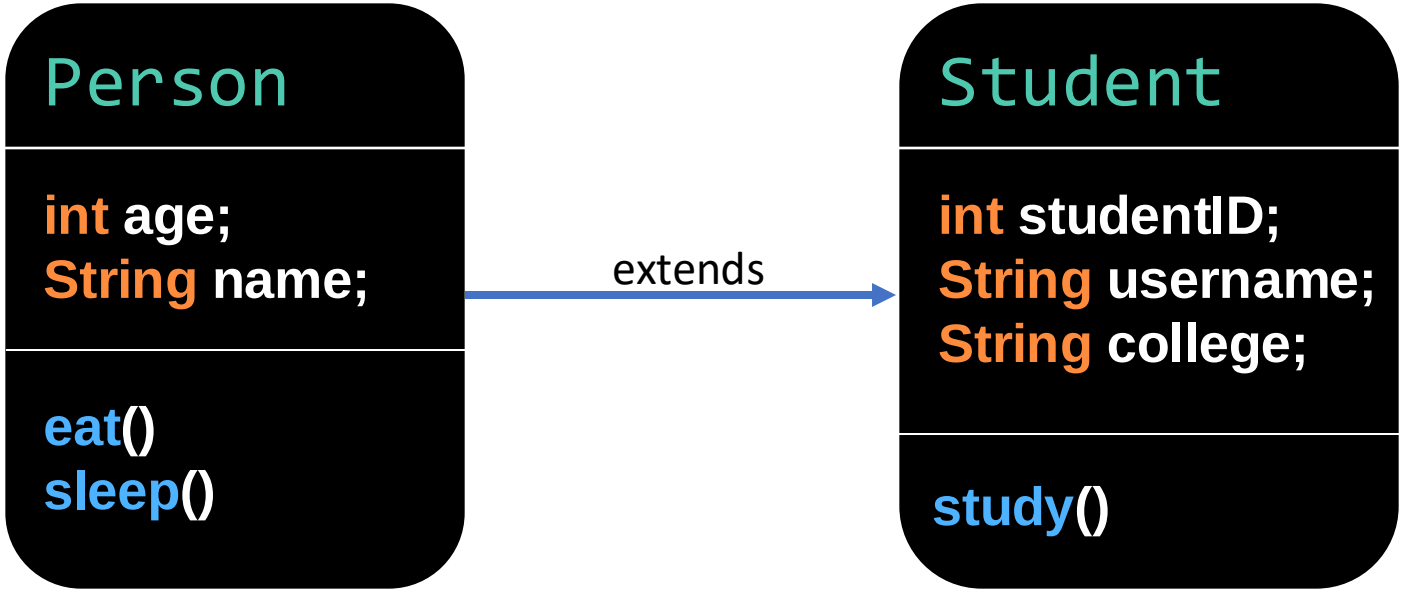

```
Person p1 = new Person("John", 35);
```

```
Person p2 = p1;
```

```
p1.setName("John Doe");
```

```
p2.getName();
```

Inheritance




```
public class Student extends Person {
    // some code
    public Student(int studentID, String username, String college, int age, String name){
        super(age,name);
        this.studentID = studentID;
        this.username = username;
        this.college = college;
    }
}
```

Overloading vs. Overriding

Lancaster University

```
public class Person {

    // some code
    public void sleep(){
        for(int h=0; h<8; h++){
            // do nothing
        }
    }

    public void eat(){
        // eat random stuff
    }
}
```

```
public class Student extends Person {

    // some code
    public void sleep(){
        // revise and do homework
        for(int h=0; h<15; h++){
            // do nothing
        }
    }

    public void eat(String meal){
        // eat chosen meal
    }
}
```

Polymorphism

Lancaster University

```
public void snooze(Person p){
  // take a break
  p.sleep(); // calls Person's sleep method
             // or Student's sleep method if p is a Student
  // wake up
}
```

```
public Person[] busPassengers = new Person[10];

// array of 10 Person objects
// each element is initialized to null
// could be Person objects or any subclass (e.g. Student)
```

Abstract classes

Lancaster University

```
public abstract class Person {

    // some code
    public void sleep(){
        for(int h=0; h<8; h++){
            // do nothing
        }
    }

    public abstract void eat();
}
```

```
Person p = new P      ();
```

ILLEGAL

```
public class Student extends Person {

    // mandatory implementation
    public void eat(String meal){
        // eat chosen meal
    }
}
```

Interfaces

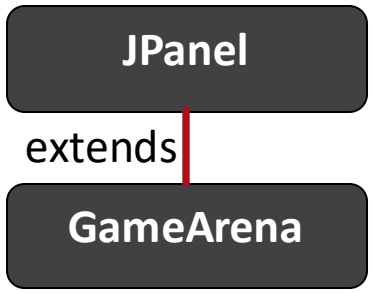
Lancaster University

```
public interface Person {

    // no attributes
    public void sleep();

    public void eat();
}
```

```
public class Student extends UM implements Person {

    // some code
    public void sleep(){
        // revise and do homework
        for(int h=0; h<15; h++){
            // do nothing
        }
    }

    public void eat(String meal){
        // eat chosen meal
    }
}
```

ERROR

Case study - GameArena



???

Ball | Line | Rectangle | Text

Summary

Lancaster University

Today we revised:

- Arrays Syntax in Java
- Core OO concepts with examples
- GameArena as a case study

Next Lecture:

- Collections