Object Oriented Analysis and Design with Java - UE19CS353

Prof. Sindhu R Pai

Department of Computer Science and Engineering

Object Oriented Analysis and Design with Java

Parameter Passing –
Value types and Reference Types

Prof. Sindhu R Pai

Department of Computer Science and Engineering

Parameter Passing - Agenda

1. Introduction

2. Value types

3. Reference types

4. Coding examples - Demo

Parameter Passing

Introduction

- Argument is copied to the parameter when some data has to be passed between methods / functions

- Parameter passing techniques   **Note: Java is Strictly Pass by Value!**

    Pass by Value

    Pass by Reference

Value types

- **Primitive types:** int, char, float etc..

- Java creates a copy of the variable being passed in the method and then do the manipulations.

- **The change is not reflected in the main method**

- If there is a change in the parameter of a value type in the called method, the variables of the caller will not change. You would have changed the copy.

```java
int data = 7;
parameter p1 = new parameter();
System.out.println(data);
p1.testing(data);
System.out.println(data);
```

```java
class parameter{
    void testing(int data)
    {        data = 89;   }
}
```

Reference types

- **Non-Primitive types are references**

- Java creates a copy of references and pass it to method, but they still point to same memory reference

- It gets tricky when we pass object references to methods

Reference types continued..

- **Trick 1: The changes are not reflected back if we change the object itself to refer some other location or object**

```java
class Test
{    int x;        Test(int i) { x = i; };        Test()        { x = 0; }
}
class Main2
{
    public static void main(String[] args)
    {    Test t = new Test(5);
        System.out.println(t.x);
        change(t);
        System.out.println(t.x);
    }
    public static void change(Test t)
    {    t = new Test();        t.x = 10;        }
}
```

- **Trick 2: Changes are reflected back if we do not assign reference to a new location or object**

```java
class Test
{    int x;        Test(int i) { x = i; };        Test()        { x = 0; }
}
class Main2
{
    public static void main(String[] args)
    {    Test t = new Test(5);
        System.out.println(t.x);
        change(t);
        System.out.println(t.x);
    }
    public static void change(Test t)
    {    t.x = 10;    }
}
```

Coding examples – Demo

THANK YOU

Prof. Sindhu R Pai

Department of Computer Science and Engineering

sindhurpai@pes.edu

+91 8277606459