

Programació

UT4.4B. Referències i Paràmetres

Referències i paràmetres

Com ja hem comentat, les variables de tipus compost així com les dels arrays no emmagatzemen directament un valor sino una referencia al lloc on es trobaran eixes dades.

També sabem que en Java els paràmetres es passen per valor (es copia l'argument en una variable local al mètode)

És important tindrè això en compte a l'hora d'utilitzar eixos tipus de dades com a paràmetres en un mètode.

Exemples

Funcionament amb paràmetres de tipus primitiu:

```
public class Parametres{
    Run | Debug
    public static void main(String[] args) {
        int quantitat1 = 2;
        int quantitat2 = 6;
        int suma = sumar(quantitat1, quantitat2);
        System.out.println(suma);
    }

    public static int sumar(int sumand1, int sumand2){
        /*
            Operacions internes quan s'invoca en la línia 5 del codi:
            sumand1 = quantitat1; sumand2 = quantitat2;
        */
        return sumand1 + sumand2;
    } // Quan finalitza la funció desapareix la variable sumand1 i sumand2
}
```

Exemples

Funcionament amb paràmetres de tipus primitiu:

```
public class Parametres{
    Run | Debug
    public static void main(String[] args) {
        double quantitat = 2.5;
        modificar(quantitat);
        // Quan finalitza el mètode, "quantitat" val 2.5
        System.out.println(quantitat);
    }

    public static void modificar(double preu){
        /*
            Operacions internes quan s'invoca en la línia 4 del codi:
            preu = quantitat;
            "preu" val 2.5 i "quantitat" val 2.5
        */
        preu = 3.75;
        // En este punt "preu" val 3.75 i "quantitat" val 2.5
    } // Quan finalitza la funció desapareix la variable preu
}
```


Exemples

Funcionament amb paràmetres de tipus array:

```
public class Parametres{
    Run | Debug
    public static void main(String[] args) {
        double[] notesTrimestrals = {7.85, 5.32, 8.5};
        System.out.println(notesTrimestrals);
        modificar(notesTrimestrals);
        // Quan finalitza el mètode, "notesTrimestrals[1]" val 9.7
        System.out.println(notesTrimestrals[0] + " " + notesTrimestrals[1] + " " + notesTrimestrals[2]);
        // Imprimirà 7.85 9.7 8.5
    }

    public static void modificar(double[] notes){
        /*
         * Operacions internes quan s'invoca en la línia 4 del codi:
         * notes = notesTrimestrals;
         * "notes" val [D@762efe5d (referència) i "notesTrimestrals" val [D@762efe5d (mateixa referència)
         */
        notes[1] = 9.7; // Està modificant la segona dada de l'array al qual apunta la referència
        // En este punt "notes[1]" val 9.7 i "notesTrimestrals[1]" val 9.7. S'ha modificat el mateix espai de memòria.
    } // Quan finalitza la funció desapareix la variable "notes"
}
```

Exemples

Este comportament és **PERILLÓS** si no es domina:

```
public class Parametres{  
    Run | Debug  
    public static void main(String[] args) {  
        double[] notesPere = {5.1, 3.2, 7.1};  
        double[] notesAitana = notesMes1(notesPere);  
        double[] notesMaria = notesMes1(notesAitana);  
  
        System.out.println(notesPere[0] + " " + notesPere[1] + " " + notesPere[2]);  
        System.out.println(notesAitana[0] + " " + notesAitana[1] + " " + notesAitana[2]);  
        System.out.println(notesMaria[0] + " " + notesMaria[1] + " " + notesMaria[2]);  
        // Els tres alumnes tenen les mateixes notes: 7.1 5.2 9.1  
    }  
  
    private static double[] notesMes1(double[] notes) {  
        notes[0]++;  
        notes[1]++;  
        notes[2]++;  
        return notes;  
    }  
}
```


Exemples

Este comportament és **PERILLÓS** si no es domina:

```
public class Parametres{
    Run | Debug
    public static void main(String[] args) {
        double[] notesTrimestrals = {7.1, 2.1, 3.6};
        modificarNotes(notesTrimestrals);
        System.out.println(notesTrimestrals[0] + " " + notesTrimestrals[1] + " " + notesTrimestrals[2]);
        // No s'han modificat les notes!!
    }

    private static void modificarNotes(double[] notes) {
        notes = new double[3];
        notes[0] = 9.5;
        notes[1] = 7.3;
        notes[2] = 7.1;
    }
}
```

Exemples

Este comportament és **PERILLÓS** si no es domina:

```
public class Parametres{
    Run | Debug
    public static void main(String[] args) {
        double[] notesTrimestrals = {7.1, 2.1, 3.6};
        modificarNotes(notesTrimestrals);
        System.out.println(notesTrimestrals[0] + " " + notesTrimestrals[1] + " " + notesTrimestrals[2]);
        // No s'han modificat les notes!!
    }

    private static void modificarNotes(double[] notes) {
        notes = new double[3];
        notes[0] = 9.5;
        notes[1] = 7.3;
        notes[2] = 7.1;
    }
}
```


Varargs (extra)

Existeix en Java la possibilitat de crear mètodes que prenguen una quantitat indeterminada d'arguments, i no un únic argument de tipus array:

```
public static void main(String[] args) {  
    int[] array1 = { 8, 5, 1, -8, 2 };  
    int resultat1 = sumarArray(array1);  
    System.out.println("sumarArray = " + resultat1);  
  
    int resultat2 = sumar(8, 5, 1, -8, 2);  
    System.out.println("sumar = " + resultat2);  
}
```

```
public static int sumarArray(int[] valors){  
    int resultat = 0;  
    for(int i=0; i<valors.length; i++){  
        resultat += valors[i];  
    }  
    return resultat;  
}  
  
public static int sumar(int ...valors){  
    int resultat = 0;  
    for(int i=0; i<valors.length; i++){  
        resultat += valors[i];  
    }  
    return resultat;  
}
```

Varargs (extra)

També es poden combinar arguments únics amb arguments de longitud variable (vararg). **El paràmetre vararg ha de ser l'últim.**

```
public class Parametres {  
    public static void main(String[] args) {  
        imprimirNotes("Pere", "Programació", 8.3, 5.1, 8.75, 7.1, 6.9);  
        imprimirNotes("Laura", "Sistemes", 9.25, 7.6);  
    }  
  
    public static void imprimirNotes(String alumne, String modul, double ...notes){  
        System.out.print(alumne + " en el mòdul de " + modul + " té les notes:");  
        for(int i=0; i<notes.length; i++){  
            System.out.print("\t" + notes[i]);  
        }  
        System.out.println();  
    }  
}
```