Initiation à la programmation

Algo avancé

PARIS – AIX-EN-PROVENCE – BORDEAUX – LILLE – LYON – NANTES – RENNES – ROUEN – SOPHIA ANTIPOLIS – STRASBOURG - TOULOUSE

Objectif

- Appréhender la récursivité
- Comprendre la notion de complexité
- Quelques algorithme classiques
- Structure de type complexe : les enregistrements

Récursivité

Exemple de la fonction “somme”

Récursivité

Exemple de la fonction “somme”

```
public void somme(int a, int b) {
    while (b > 0) {
        a++;
        b--;
    }
}
```

Récursivité

Exemple de la fonction “somme”

```
public void somme(int a, int b) {
    while (b > 0) {
        a++;
        b--;
    }
}
```

```
public void sommeRec(int a, int b) {
    if (b== 0) {
        return a;
    } else {
        return sommeRec(a+1, b-1);
    }
}
```

Récursivité

La récursivité est particulièrement utilisée dans le cas de l'IA.

Les échecs, Zyan Drench

Notion de complexité

La complexité d’un algorithme est le calcul des ressources qu’il consomme.

Notion de complexité

La complexité d'un algorithme est le calcul des ressources qu'il consomme.

On distingue la complexité spatiale, qui mesure le nombre d'emplacements mémoires requis par l'algorithme, de la complexité temporelle, qui mesure le temps d'exécution d'un algorithme.

Notion de complexité

Exemple de l'affichage d'un tableau

```
int[] array = new int[] {1, 2, 5, 7, 4, 6, 9};

for (int i = 0; i < array.length; i++) {
    System.out.println(tableau[i]);
}
```

Notion de complexité

Exemple de l'affichage d'un tableau

```
int[] array = new int[] {1, 2, 5, 7, 4, 6, 9};

for (int i = 0; i < array.length; i++) {
    System.out.println(tableau[i]);
}
```

O(7)

Notion de complexité

Exemple de l'affichage d'un tableau

```
int[] array = new int[] {1, … , 9};

for (int i = 0; i < array.length; i++) {
    System.out.println(tableau[i]);
}
```

Notion de complexité

Exemple de l'affichage d'un tableau

```
int[] array = new int[] {1, … , 9};

for (int i = 0; i < array.length; i++) {
    System.out.println(tableau[i]);
}
```

O(N)

Notion de complexité

Quelques complexités courantes…

O(log(N))	O(N)	O(N*log(N))	O(n*m)	$O(n^2)$

Nous en croiserons quelques-unes dans les exercices qui viennent ☺

Notion de complexité

Quelques algos connus

La recherche séquentielle
La recherche dichotomique
La recherche aléatoire

Calcul factoriel

Le tri à bulle
Le tri stupide
Le tri rapide

Les types complexes

Les types complexes

Les **types complexes** désignent les types capables de stocker de multiples informations **fortement liées entre elles**.

Les types complexes

Les types complexes

Les **types complexes** désignent les types capables de stocker de multiples informations **fortement liées entre elles**.

Selon le contexte (langage, algo), on leur donne différents noms.

Les types complexes

Les types complexes

Les **types complexes** désignent les types capables de stocker de multiples informations **fortement liées entre elles**.

Selon le contexte (langage, algo), on leur donne différents noms.

En algo, on les appelle “**enregistrements**”.
En Java, nous les appellerons “**classes**”.

Les types complexes

Nous les écrirons ainsi :

```
public class Personne {
    public int age;
    public String nom;
}
```

Vous pouvez le comprendre comme la création d'**un nouveau type personnalisé**.

Les types complexes

```
public class Personne {
    public int age;
    public String nom;
}
```

```
Personne quelquun;
quelquun.age = 42;
quelquun.nom = Polnareff;
System.out.println(quelquun.age);
System.out.println(quelquun.nom);
```

Atelier 8
Algo avancé

Initiation à la programmation

Algo avancé

PARIS – AIX-EN-PROVENCE – BORDEAUX – LILLE – LYON – NANTES – RENNES – ROUEN – SOPHIA ANTIPOLIS – STRASBOURG - TOULOUSE