IFT 1015 - Sujets divers

Professeur:

Stefan Monnier

B. Kégl, S. Roy, F. Duranleau, S. Monnier

Département d'informatique et de recherche opérationnelle

Université de Montréal

hiver 2006

Au programme

Opérateurs, effet de bord

Évaluation “paresseuse” de `&&` et `||`

Opérateur de sélection

Analyser des boucles

Implanter des boucles

Opérateurs

Règles

- un opérateur sort toujours une valeur
- un opérateur peut changer un de ses paramètres: effet de bord

Exemples d'effet de bord: $=$ $+=$ ... $++$ $--$

- les valeurs sorties de ces opérateurs sont utilisées rarement

Deux types d'opérandes: référence et valeur

Évaluation "paresseuse" de `&&` et `||`

L'évaluation arrête quand la condition est déterminée

```
int i = 3, n = 2;
if (i == 3 || n++ < 2) ...;
if (i < 2 && (n = 4) < 8) ...;
if (n++ < 2 || i == 3) ...;
if ((n = 4) < 8 && i < 2) ...;
```

- ÉVITER les programmes comme ça!!!
- plus précisément: éviter les opérateurs avec un effet de bord dans les expressions conditionnelles

Évaluation "paresseuse" de `&&` et `||`

Parfois utile

```
if (input != null
    && Integer.parseInt(input) > 0)
   ...
```

Pour assurer l'évaluation de tous les opérandes

- utiliser les opérateurs `&` et `|`

L'opérateur de sélection

Syntaxe: `<test> ? <exp1> : <exp2>`

- si `<test>` est vrai, `<exp1>` est retournée, sinon, `<exp2>` est retournée

Exemple:

```
y = x >= 0 ? x : -x;
```

équivalent à

```
if (x >= 0)
   y = x;
else
   y = -x;
```

Analyse des boucles

```
int methode(int valeur) {
    int res = 0;
    while (valeur != 0) {
        if (((valeur % 10) % 2) == 1)
            ++res;
        valeur /= 10;
    }
    return res;
}
```

Analyse des boucles

```
void methode (int n) {
   if ((n % 2) == 1) && n > 2) {
      int mil = n / 2;
      for (int i=0; i < n; i++) {
         for (int j=0; j < n; j++)
            System.out.print(
              ((j == mil) || (i == mil))
              ? "*" : " ");
         System.out.println();
      }
   }
}
```

Analyse des boucles

```
void methode (int n) {
    int m = 3 * n / 4;
    int p = 2 * n - m;
    for (int i = 0; i < n; i++) {
        int j1, j2, j3;
        for (j1 = 0; j1 < m; j1++)
            System.out.print("*");
        for (j2 = j1; j2 < n && j2 < p; j2++)
            System.out.print(" ");
        for (j3 = j2; j3 < n; j3++)
            System.out.print("*");
        System.out.println();
        m--; p--;
    }
}
```

Implantation des boucles

- Qu'est-ce qu'il faut faire dans chaque itération?
- Combien de fois est-ce qu'il faut répéter la boucle?
  - nombre défini: boucle de `for`
  - nombre indéfini: boucle de `while`
- Mettre les opérations d'étape 1 dans le corps de la boucle
- Vérifier les initialisations
- Vérifier les erreurs ± 1

Implantation des boucles

- H2002/6
- H2000/6
- A2001/6
- A2001/7