

IFT 1015 - Art de programmer

Professeur:
Stefan Monnier

B. Kégl, S. Roy, F. Duranleau, S. Monnier
Département d'informatique et de recherche opérationnelle
Université de Montréal

hiver 2006

Premier d'une série sur l'art de bien programmer.

- Arrondir les nombres
- Les commentaires
- Vérification et traces d'exécution

Types d'arrondi

- On arrondit le nombre réel r vers l'entier e
 - **plancher**: plus grand entier e qui satisfait $e \leq r$
exemples: $1.1 \rightarrow 1$, $1.8 \rightarrow 1$
 - **arrondi**: entier e le plus près de r
exemples: $1.1 \rightarrow 1$, $1.8 \rightarrow 2$, $1.5 \rightarrow 2$
 - **plafond**: plus petit entier e qui satisfait $e \geq r$.
exemples: $1.1 \rightarrow 2$, $1.8 \rightarrow 2$, $1 \rightarrow 1$

Plancher

```
e = (int)Math.floor(r);
```

```
e = (int)r;
```

Arrondi

```
e = Math.round(r);
```

Plafond

```
e = (int)Math.ceil(r);
```

- si r est rationnel: $r = n/d$, n et d sont entiers

```
e = (n+d-1)/d;
```

Exemples

- Combien d'avions de 135 places faut-il pour transporter 505 personnes? → Plafond de $505/135$

```
e = (int)(505+135-1)/135; // = 4
```

- Combien faut-il d'aquarium de 100 litres pour conserver 234 litres d'eau? → Plafond de $234/100$

```
e = (int)(234+100-1)/100; // = 3
```

- Quand commenter?:

Lorsque quelqu'un **pourrait ne pas comprendre** le sens d'une instruction.

- doit décrire le **sens**, pas l'instruction
- ne pas commenter l'évidence

Les commentaires

Exemples (séparez le bon du mauvais)

- `i = i + 10; // Passe au prochain pixel de l'image`
- `i = i + 10; // Ajoute dix à la position dans l'image`
- `i = i + 10; // Ajoute dix à la variable i`
- `i = i + 10; // Modifie la position dans l'image`

Les commentaires

Quoi commenter

- les **déclarations** (de variables, de fonctions, ...)
- les **opérations** du programme
- les **algorithmes**

Exemples

- `int i; // Position horizontale du pixel dans l'image`
`...`
`i = i + 10; // Passe au prochain pixel`
- `double c; // Cout d'un colis (en sous)`
`...`
`c = c * 1.07; // Ajoute la TPS au cout du produit`

Self-commenting code

- ```
int posX;
final int PIXEL_STEP_SIZE = 10;
...
posX += PIXEL_STEP_SIZE;
```
- ```
double coutColis;  
final double TAUX_TPS = 0.07;  
...  
coutColis += coutColis * TAUX_TPS;
```

Vérification et traces d'exécution

Comment savoir si mon programme fonctionne?

- **imprimer** des informations sur le déroulement du programme
- vérifier chaque élément du programme **séparément**

```
double cout = 31.45;
double tvq
    = Math.round
      ((cout
        + Math.round(0.07 * cout * 100) / 100.0)
       * 0.08 * 100)
    / 100.0;
```

Vérification et traces d'exécution

```
double cout = 31.45;  
final double TAUX_TPS = 0.07;  
final double TAUX_TVQ = 0.08;  
  
System.out.println("cout = " + cout);
```

- sortie au terminal

```
cout = 31.45
```

Vérification et traces d'exécution

```
double tps = TAUX_TPS * cout;  
System.out.println("avant arrondi: tps = cout * "  
    + TAUX_TPS + " = " + tps);
```

```
tps = Math.round(tps * 100) / 100.0; // arrondi  
System.out.println("apres arrondi: tps = cout * "  
    + TAUX_TPS + " = " + tps);
```

- sortie au terminal

```
avant arrondi: tps = cout * 0.07 = 2.20150000000000  
apres arrondi: tps = cout * 0.07 = 2.2
```

Vérification et traces d'exécution

```
double tvq = TAUX_TVQ * (cout + tps);  
System.out.println("avant arrondi: tvq = (cout + tps) * "  
    + TAUX_TVQ + " = " + tvq);
```

```
tvq = Math.round(tvq * 100) / 100.0; // arrondi  
System.out.println("apres arrondi: tvq = (cout + tps) * "  
    + TAUX_TVQ + " = " + tvq);
```

sortie au terminal

```
avant arrondi: tvq = (cout + tps) * 0.08 = 2.6919999999999999  
apres arrondi: tvq = (cout + tps) * 0.08 = 2.69
```

Vérification et traces d'exécution

Après le test: effacer ou "comment out"

```
double tvq = TAUX_TVQ * (cout + tps);  
//System.out.println("avant arrondi: tvq = (cout + tps) * "  
//      + TAUX_TVQ + " = " + tvq);  
  
tvq = Math.round(tvq * 100) / 100.0; // arrondi  
//System.out.println("apres arrondi: tvq = (cout + tps) * "  
//      + TAUX_TVQ + " = " + tvq);
```

Emacs

- **comment out** du code: **marquer** le code et **M-;**
- **indentation** correcte: **TAB**

Réfléchir avant coder

Ne pas tout coder d'un seul coup

- Nombre d'erreurs
= (Nombre de lignes de code)
+ (Nombre de lignes de code non testées)²
- Nombre d'erreurs restantes = Nombre d'erreurs corrigées