Langages objets

Fonctions

M2 Pro CCI, Informatique
Emmanuel Waller, LRI, Orsay

résumé des épisodes précédents

- découverte Java, prise en main environnement, types primitifs
- opérateurs et expressions, instructions de contrôle, débugage
- tableaux

Les fonctions

- Exemples
- Syntaxe
- Portée des variables
- Passage des paramètres et modification
- Déroulement mémoire
- Allocation dynamique dans une fonction
- Motivation

Exemple

- Écrire et utiliser une fonction qui prend en entrée un entier n et renvoie la valeur n + 1
- Démonstration (Ex1)

Exemple

- Écrire et utiliser une fonction qui prend en entrée deux entiers n et m et affiche la valeur n + m
- Démonstration (Ex2)

Syntaxe

```
static type-retour nom-fonction (paramètres) {
        // cette ligne est la signature de la fonction
    [ instructions ]
        // y compris déclarations de variables
    return résultat;   // obligatoire (sauf subtilités)
}
```

- Types paramètres et retour quelconques (primitifs, tableaux, etc.)
- Cas où pas de type de retour
  - la fonction ne renvoie rien, c'est une « procédure »
  - on indique void en type de retour
  - Le return (sans paramètre) est facultatif
- Static : comme main (admis pour l'instant)

Exemple

- Afficher un tableau
- Démonstration (Ex3)

Les fonctions

- Exemples
- Syntaxe
- Portée des variables
- Passage des paramètres et modification
- Déroulement mémoire
- Allocation dynamique dans une fonction
- Motivation

Portée d'une variable

- visibilité, « utilisabilité »
- de sa déclaration à la première accolade fermante qui la suit (bloc, corps de boucle, fonction)
  - Même les fonctions appelées dans ce bloc n'y ont pas accès
- Si erreur de portée : échec compilation : message d'erreur
- Portée définie par le texte du programme (ne change pas d'une exécution à l'autre)

exemple

- Démonstration (Ex5)

Modifier une variable dans une fonction ?

- Une fonction ne peut pas modifier la valeur d'une variable d'un type primitif extérieure à elle-même
- On dit que les arguments sont passés par valeur (c'est-à-dire recopie)

exemple

- Démonstration (Ex6)

Cases

Modifier un tableau passé en argument

- Une fonction peut modifier les valeurs d'un tableau extérieur à elle-même
- On dit que les tableaux sont passés par référence (ou par variable, ou par adresse)
- En effet, c'est la référence au tableau (son adresse) qui est passée à la fonction
- Bien sûr, cette adresse est passée par... valeur (recopiée)

exemple

- Démonstration (Ex7)

Cases

Les fonctions

- Exemples
- Syntaxe
- Portée des variables
- Passage des paramètres et modification
- Déroulement mémoire
- Allocation dynamique dans une fonction
- Motivation

déroulement mémoire : appel et retour d'une fonction lors appel f(x)

1. allouer zone paramètres formels + zone return
2. copier les valeurs des paramètres actuels dans la zone des formels

3. exécution de la fonction

- avec accès aux variables
  - siennes propres (locales)
  - celles dans la portée desquelles elle se trouve (donc globales à la classe : pas dans CCI-LO)

  et aux zones (tableaux et objets) référencées
- comme si c'était main (qui est le cas de base : pas d'appelant)

4. lors return : valeur de retour recopiée dans zone return (sauf si void)
5. si valeur de retour utilisée dans affectation dans fonction appelante : effectuer l'affectation avec le contenu de la case return
6. les zones mémoire des variables (donc hors zones référencées) de f sont libérées

Les fonctions

- Exemples
- Syntaxe
- Portée des variables
- Passage des paramètres et modification
- Déroulement mémoire
- Allocation dynamique dans une fonction
- Motivation

Allocation dynamique dans une fonction

- ex : créer un tableau de taille donnée dans une fonction, et renvoyer son adresse
- Remarque : si

  void init(int [ ] t)

  il faut créer t hors de init : sale, éviter

  - Création et initialisation doivent se faire dans une fonction (modularité)

exemple

- Démonstration (Ex8)

Cases

exemple

- saisie
- Démonstration (Ex9)

Les fonctions

- Exemples
- Syntaxe
- Portée des variables
- Passage des paramètres et modification
- Déroulement mémoire
- Allocation dynamique dans une fonction
- Motivation

Quand (et pourquoi) écrire une fonction ?

- On veut réutiliser un morceau de code (factoriser)
  - Ex : deux calculs de factorielle
- On a identifié un morceau de code ayant une certaine cohérence (clarifier)
  - Ex : un calcul de factorielle
- Si le code d'une fonction dépasse une page, on en fait une deuxième (clarifier)

Les fonctions : récapitulatif

- Exemples
- Syntaxe
- Portée des variables
- Passage des paramètres et modification
  - Types primitifs par valeur
  - Tableaux (et objets) par référence (Delannoy chap. 7.3)
- Déroulement mémoire d'une fonction

- Allocation dynamique d'un tableau (ou d'un objet) dans une fonction
- Motivation (à quoi ça sert ?)
- Delannoy chap. 6.7.2

délégués ?