Support de cours

Cours:

Initiation à la programmation (en C++)

Vidéo:

Définitions (partie 1)

Concepts (extraits des sous-titres générés automatiquement) :

Facettes des fonctions. Expressions return. Appel d'une fonction. Exemple du calcul de la moyenne. Ensemble des instructions. Corps de la fonction. Notion de définition. Prototype d'une fonction. Ensemble d'instructions c. Définition de la fonction moyenne. Simple instruction return. Séquences précédentes. Seule instruction return. Définition d'une fonction. Sens c.

vers la recherche de séquences vidéo
(dans Initiation à la programmation (en C++).)

vers la vidéo

EPFL

Fonctions : définitions
(Partie 1)

Initiation à la programmation (C++)

Vincent Lepetit, Jean-Cédric Chappelier et Jamila Sam

...

notes

résumé

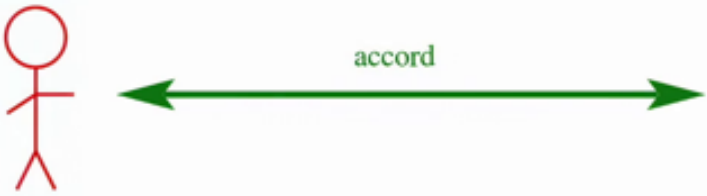
0m 0s

Les « 3 facettes » d'une fonction

- Résumé / Contrat (« prototype »)
- **Création / Construction (« définition »)**
- Utilisation (« appel »)

programmeur
utilisateur

programmeur
concepteur/développeur



appel

```
z = f(2*u, v+3);
```

prototype

```
double f(double x,
         double y);
```

définition

```
double f(double a, double b)
{
  ...
}
```

Dans les séquences précédentes, nous avions vu ce que sont les trois facettes des fonctions.

notes

résumé

0m 1s

Exemple complet

```
#include <iostream>
using namespace std;

double moyenne(double nombre_1, double nombre_2);

int main()
{
  double note1(0.0), note2(0.0);
  cout << "Entrez vos deux notes : " << endl;
  cin >> note1 >> note2;
  cout << "Votre moyenne est : "
       << moyenne(note1, note2) << endl;
  return 0;
}

double moyenne(double x, double y)
{
   return (x + y) / 2.0;
}
```

Nous avions vu ce qu'est l'appel d'une fonction, nous avions vu aussi dans une autre séquence ce qu'est le prototype d'une fonction. Dans cette séquence-ci, nous allons plus particulièrement étudier la notion de définition. Si je reprends l'exemple du calcul de la moyenne,

notes

résumé

0m 9s

la définition de la fonction moyenne, ce sont ces lignes ici en bas du programme
Nous allons maintenant nous focaliser plus particulièrement là-dessus.

notes

résumé

0m 25s

Définition des fonctions

définition d'une fonction :

- spécification du **corps** de la fonction

Syntaxe :

```
type nom ( liste de paramètres )
{
    instructions du corps de la fonction;
    return expression;
}
```

Exemple :

```
double moyenne (double x, double y)
{
   return (x + y) / 2.0;
}
```

La définition d'une fonction sert, comme son nom l'indique, à définir une fonction techniquement,

notes

résumé

0m 36s

Définition des fonctions

définition d'une fonction :

- spécification du **corps** de la fonction

Syntaxe :

```
type nom ( liste de paramètres )
{
    instructions du corps de la fonction;
    return expression;
}
```

Exemple :

```
double moyenne (double x, double y)
{
   return (x + y) / 2.0;
}
```

c'est-à-dire à spécifier le corps de la fonction, c'est-à-dire l'ensemble des instructions qui ont fait qu'on a choisi de faire cette fonction. Plus particulièrement, la définition d'une fonction va commencer par la donnée de l'entête de la fonction, c'est-à-dire le début du prototype sans le point virgule où je vous rappelle que l'on définit d'abord le type, puis le nom de la fonction et enfin, entre parenthèses rondes, l'ensemble des paramètres que l'on souhaite fournir à la fonction. Pour la définition, derrière cet entête, vient ensuite le bloc qui est le corps de la fonction. Ce corps est donc un ensemble d'instructions C++ usuel que l'on a mis dans un bloc ici

notes

résumé

0m 42s

Corps de fonction

Le corps de la fonction est donc un **bloc**
dans lequel on peut utiliser les paramètres de la fonction.

La valeur retournée par la fonction est indiquée par l'instruction :

```
return expression;
```

où l'*expression* a le même *type* que celui retourné par la fonction.

```
double moyenne (double x, double y)
{
   return (x + y) / 2.0;
}
```

L'instruction `return` fait deux choses:

- elle précise la valeur qui sera fournie par la fonction en résultat
- elle met fin à l'exécution des instructions de la fonction.

L'expression après `return` est parfois réduite à une seule variable ou même à une valeur littérale, mais ce n'est pas une nécessité.

et qui contient une ou plusieurs expressions return qui mettront fin à l'exécution du corps de la fonction. Si je prends un exemple, l'exemple de la fonction moyenne, on retrouve ici l'entête de la fonction avec le type de retour double, la moyenne va retourner un double, son nom et la liste des deux arguments. Ici, on va faire la moyenne entre deux nombres x et y. Dans la définition, on a le corps ici de la fonction qui est ce bloc qui est ici réduit à une simple instruction return sans +. Le corps de la fonction est simplement un bloc au sens C++, c'est-à-dire un ensemble d'instructions C++ comprise entre une accolade ouvrante et une accolade fermante. La seule différente est que l'on dispose de variables supplémentaires dans ce bloc qui sont les paramètres de la fonction que l'on peut utiliser dans ce bloc comme n'importe quelle variable usuelle. La valeur retournée par la fonction est indiquée par l'instruction return. Cette instruction est suivie par une expression au sens C++ qui sera évaluée et qui donnera la valeur que la fonction va retourner. Cette expression doit être de même type que le type de retour de la fonction. Dans l'exemple du calcul de la moyenne, nous avons ici le corps de la fonction qui est ce bloc ici, qui est très simple dans ce cas-là qui ne contient qu'une seule instruction return laquelle va donc évaluer l'expression qui est ici où vous voyez que l'on récupère et que l'on peut utiliser les deux paramètres x et y. L'expression sera évaluée, on va en faire x + y puis, on va diviser par deux et le résultat qui est de type double sera renvoyé par la fonction au reste du programme. L'instruction return fait deux choses : elle permet de préciser la valeur de retour qui sera fournie

notes

résumé

1m 25s

Corps de fonction

Le corps de la fonction est donc un **bloc**
dans lequel on peut utiliser les paramètres de la fonction.

La valeur retournée par la fonction est indiquée par l'instruction :

```
return expression;
```

où l'*expression* a le même *type* que celui retourné par la fonction.

```
double moyenne (double x, double y)
{
   return (x + y) / 2.0;
}
```

L'instruction `return` fait deux choses:

- elle précise la valeur qui sera fournie par la fonction en résultat
- elle met fin à l'exécution des instructions de la fonction.

L'expression après `return` est parfois réduite à une seule variable ou même à une valeur littérale, mais ce n'est pas une nécessité.

par la fonction au reste du programme et elle termine l'exécution du corps de la fonction dès le premier return rencontré, la fonction s'arrête et retourne la valeur correspondant à l'expression.

notes

résumé

Très souvent, vous trouverez dans du code des expressions return relativement simples qui se réduit simplement à un nom de variables comme par exemple, return x où x ici est un nom de variables ou même des valeurs par exemple, return 3 pour renvoyer la valeur trois. Mais ceci est un cas tout à fait particulier on peut absolument, et comme on l'a vu précédemment, utiliser une expression complète au sens C++ et c'est ça qui est la cas général d'évaluer une expression complète. x et 3 ici sont simplement des expressions particulières. des expressions particulières.

notes

résumé

3m 37s