

Support de cours

Cours:

Introduction à la programmation orientée objet (en C++)

Vidéo:

W17-03-surcharge-CPP-pt2

Concepts (extraits des sous-titres générés automatiquement) :

Méthode prix de la montre. Programme exécutable. Opérateur d'affichage. Exécution de ce programme. Notion de copie d'une montre. Résultat suivant. Séquences vidéo suivantes. Différents accessoires. Lequel opérateur d'affichage. Héritage multiple. Prix total. Affichage du message prix total. Appel de la méthode. Méthode prix. Somme des prix des différents.



[vers la recherche de séquences vidéo](#)

(dans Introduction à la programmation orientée objet (en C++).)



[vers la vidéo](#)

Center for Digital Education. Plus de matériel de soutien pédagogique ici :

<https://www.epfl.ch/education/educational-initiatives/cede/educational-technologies-gallery/boocs-en/>

Etude de cas : surcharge d'opérateur et première version

(Partie 2)

Introduction à la programmation orientée objet (en C++)

Jean-Cédric Chappelier, Jamila Sam et Vincent Lepetit

...

notes

résumé

0m 0s



Pour finaliser la classe `Montre` (mais sans mécanisme) :

- ▶ ajoutons un constructeur (au moins remettre le constructeur par défaut)
- ▶ décidons d'un calcul de prix : somme des prix des accessoires
- ▶ décidons d'un affichage :

Une montre composée de :
 * bracelet cuir coûtant 54
 * fermoir acier coûtant 12.5
 * etc.
 ==> Prix total : 147.9

```
// ...
virtual void afficher(ostream& sortie)
    const override {

    sortie << "Une montre composée de : "
        << endl;

    for (auto const& p_acc : accessoires) {
        sortie << " * " << *p_acc << endl;
    }

    sortie << "==> Prix total : " << prix()
        << endl;
}
// ...
```

Oui, c'est correct et ça compile. Et enfin on termine par l'affichage du message prix total, et l'appel ici à la méthode `prix` de la montre,

notes

résumé

0m 1s





qui va donc nous calculer le prix total, comme expliqué précédemment. Enfin, pour avoir un programme exécutable, il faut bien un « main », que nous allons donc ici utiliser pour tester

notes

résumé

0m 13s



Un exemple de main()

```
int main() {

    Montre m;

    m += new Bracelet("cuir", 54.0);
    m += new Fermeoir("acier", 12.5);
    m += new Boitier("acier", 36.60);
    m += new Vitre("quartz", 44.80);

    cout << "Montre m :" << endl;
    cout << m << endl;

    return 0;
}
```

```
Montre m :
Une montre composée de :
    * bracelet cuir coutant 54
    * fermeoir acier coutant 12.5
    * boitier acier coutant 36.6
    * vitre quartz coutant 44.8
==> Prix total : 147.9
```

Le code complet à ce stade (134 lignes) peut être téléchargé sur le site du cours :

<https://d396qusza40orc.cloudfront.net/cpp-fr/complements/final01.cc>

les implémentations des objets que nous venons de créer. Pour ceci, dans le « main », nous allons déclarer une montre, nous pouvons tout à fait faire ceci puisque nous avons un constructeur par défaut à ce stade là. Ensuite, dans la montre, nous ajoutons différents accessoires, comme ceci, chacun des accessoires a son constructeur avec le nom ou le complément du nom, et son prix. Donc on ajoute par exemple, comme ça, 4 accessoires, et enfin on affiche un message pour afficher la montre ici, avec l'opérateur d'affichage que nous avons vu précédemment, lequel opérateur d'affichage, nous venons de le voir, va appeler la méthode prix. L'exécution de ce programme va donc conduire au résultat suivant, on va d'abord afficher le message « Montre m : », puis ensuite on va rentrer dans l'opérateur surchargé pour les produits, mais avec un appel de la méthode affichée polymorphique, pour les montres, qui va donc nous afficher « Une montre composée de : », et ensuite une boucle sur ses différents accessoires, et puis qui termine par l'affichage du prix total,

notes

résumé

0m 25s





somme des prix des différents accessoires. Pour ceux qui le souhaitent, l'entièreté du code est disponible sur le site du cours. Ceci termine la présentation de cette première version opérationnelle de notre code. Reste encore 2 chapitres à aborder dans les 2 séquences vidéo suivantes, tout d'abord tous ces mécanismes, avec l'héritage multiple, et puis enfin la notion de copie d'une montre, comment faire une copie profonde des différents accessoires d'une montre. des différents accessoires d'une montre.

notes

résumé

1m 25s

