

Examen de C++

Durée : 1h00
Aucun document autorisé
Mobiles interdits

- 1. Citez et décrivez 4 qualités fondamentales recherchées dans la conception et le développement d'un logiciel.

voir cours

- 2. En programmation objet, quels sont les rôles des procédures et des fonctions associées à un objet ?
-

voir cours

- 3. Quel est, le ou les, modes de transmission des paramètres en C++ ?
-

voir cours

On souhaite représenter (de façon très simplifiée) le stock d'un magasin de jouets.

- 4. Dans le fichier `jouet.hpp`, déclarez la classe `jouet` qui contient :
- les trois variables membres `nom`, `quantité`, `prix`, respectivement de type `std::string`, `int` et `double`;
 - le prototype du constructeur pour initialiser les variables membres, plus celui du constructeur par défaut ;
 - les prototypes des méthodes `accesseur getNom`, `getQuantité`, `getPrix`;
 - le prototype de la méthode `approvisionner` qui ajoute `n` pièces à la quantité du jouet courant ;
 - le prototype de la méthode `toString` qui renvoie la représentation sous forme d'une chaîne de caractères de l'objet courant.
-

```
class jouet {  
private:  
    std::string nom;  
    int quantité;  
    double prix;  
public:
```

```
// les constructeurs  
jouet(const std::string &n, const int q, const double p);  
jouet();  
// les accesseurs  
std::string getNom() const;  
int getQuantité() const;  
double getPrix() const;  
//  
void approvisionner(const int n);  
std::string toString() const;  
};  
.....
```

- 5. Dans un fichier `jouet.cpp`, donnez la programmation des deux constructeurs.
-

```
jouet::jouet() {};  
  
jouet::jouet(const std::string &n, const int q, const double p) :  
    nom(n), quantité(q), prix(p) {}  
.....
```

- 6. Dans un fichier `jouet.cpp`, donnez la programmation des méthodes `getPrix` et `approvisionner`.
-

```
/*  
 * Rôle : renvoie le prix du jouet courant  
 */  
double jouet::getPrix() const { return this->prix; }  
  
/*  
 * Antécédent : n>0  
 * Rôle : augmente la quantité du jouet courant de n pièces  
 */  
void jouet::approvisionner(const int n) {  
    assert(n>0);  
    this->quantité+=n;  
}  
.....
```

Le fichier `stock.hpp` contient la déclaration de la classe `stock` suivante :

```
class stock {  
private:  
    std::vector<jouet> leStock; // contient tous les jouets du stock  
public:  
    stock(const std::string &f);  
    double valeurStock() const;  
};
```

- 7. Dans le fichier `stock.cpp`, écrivez la méthode `valeurStock` qui renvoie la valeur totale (en euros) du stock.

```
/**  
 * rôle : renvoie la valeur du stock courant  
 */  
double stock::valeurStock() const {  
    double valeurTotale=0.0;  
    for (const jouet &p : this->leStock)  
        valeurTotale+=p.getPrix()*p.getQuantité();  
    //  
    return valeurTotale;  
}
```

.....

Un fichier de texte contient une liste de jouets, avec sur chaque ligne : son *nom*, sa *quantité* et son *prix*. Par exemple :

```
ballon 100 9.99  
monopoly 10 150.0  
puzzle 150 30.5  
...
```

On considère aussi que la classe *jouet* possède l'opérateur `>>` pour lire les 3 composants d'un jouet dans un *ifstream*.

```
friend std::ifstream & operator>>(std::ifstream &f, jouet &j);
```

- 8. Dans le fichier *stock.cpp*, écrivez le constructeur qui initialise le stock à partir d'un fichier défini comme précédemment.

```
/*  
 * antécédent : f fichier de jouets  
 * rôle : initialise le stock à partir du fichier f  
 */  
stock::stock(const std::string &f) {  
    std::ifstream is(f);  
    // vérifier si le fichier est correctement ouvert en lecture  
    if (!is.is_open()) {  
        std::cerr << f.c_str();  
        std::exit(EXIT_FAILURE);  
    }  
    // le fichier est ouvert sans erreur => remplir le vecteur leStock  
    jouet j;  
    while (is >> j)  
        this->leStock.push_back(j);  
    //  
    is.close();  
}
```

.....