Université de Nice-Sophia Antipolis — POLYTECH
ELEC4 — 2021–2022

Examen de C++ - Conception Objet

Durée : 1h
Aucun document autorisé
Mobiles interdits

▶ 1. Citez et expliquez 4 qualités fondamentales que doit viser un logiciel informatique dans sa conception OO.

---

voir cours

.....

▶ 2. Expliquez ce qu'est un *patron de conception*, puis expliquez le rôle, l'intérêt... du patron *Itérateur*.

---

voir cours

.....

On possède la classe `vin` suivante pour représenter les vins de la cave d'un œnologue amateur.

```
enum t_couleur { ROUGE, ROSE, BLANC };

class vin {
private:
  std::string nom;
  int annee;
  enum t_couleur couleur;
  std::string region;

public:
  vin(std::string n, enum t_couleur c, std::string r, int a);
  bool operator==(vin v);
  std::string getNom() const;
  t_couleur getCouleur() const;
  std::string getRegion() const;
  int getAnnee() const;
  const std::string toString() const;
  friend std::ostream &operator<< (std::ostream &f, const vin &p);
};
```

▶ 3. La cave est représentée par une liste de vins. Écrivez la déclaration de classe `listeDeVins` par héritage du type générique `std::list`. Pour l'instant, cette classe est vide.

```
class listeDeVins : public std::list<vin> {
};
```

.....

L'œnologue amateur veut appliquer sur sa liste de vins des filtres pour extraire des vins selon des critères spécifiques.

▶ 4. Écrivez la classe abstraite générique `filtre` qui possède la méthode virtuelle pure `predicat` qui prend en paramètre *donnée* un objet de type générique et renvoie un booléen.

```
template <typename T>
class filtre {
public:
  virtual bool predicat(const T &x) const =0;
};
```

.....

▶ 5. Par héritage, écrivez les 3 classes `filtreRouge`, `filtreBordeaux` et `filtre2002`, qui implémentent chacune la méthode `predicat` pour filtrer respectivement les vins rouges, de bordeaux et de l'année 2002.

```
class filtreRouge : public filtre<vin> {
  virtual bool predicat(const vin &x) const override {
    return x.getCouleur()==ROUGE;
  }
};

class filtreBordeaux : public filtre<vin> {
  virtual bool predicat(const vin &x) const override {
    return x.getRegion() == "Bordeaux";
  }
};


class filtre2002 : public filtre<vin> {
  virtual bool predicat(const vin &x) const override {
    return x.getAnnee()==2002;
  }
};
```

.....

▶ 6. Dans la classe `listeDeVins`, écrivez la méthode `appliquerFiltre` qui applique sur une liste de vins un filtre et renvoie la liste filtrée. Son en-tête est le suivant :

```
listeDeVins appliquerFiltre(const listeDeVins &l, const filtre<vin> *f) {
```

```
listeDeVins appliquerFiltre(const listeDeVins &l, const filtre<vin> *f) {
  // appliquer le filtre
  listeDeVins _l;
```

```
    for (vin x : l) {
      if (f->predicat(x))
        _l.push_front(x);
    }
    return _l;
  }
```

.....

Dans la méthode `main`, on possède une liste `l` des vins suivants :

```
"Château Margaux", ROUGE, "Bordeaux", 1982
"Château de Parenchère", ROSE, "Bordeaux", 2022
"Château Margaux", ROUGE, "Bordeaux", 2002
"Mouton Cadet", BLANC, "Bordeaux", 2017
"Château Barbanau", BLANC,"Cassis", 2002
"Sainte Croix", ROSE, "Côte de Provence", 2021
"Montrachet", ROUGE, "Bourgogne", 1998
"Château de Meursault", BLANC, "Bourgogne", 2018
```

► 7. Écrivez le code pour filtrer les vins rouges, puis les vins de bordeaux.

```
    l.appliquerFiltre(l, new filtreRouge());
    l.appliquerFiltre(l, new filtreBordeaux());
```

.....

► 8. On veut maintenant pouvoir faire des opérations logiques sur les filtres. Ajoutez à la classe `listeDeVins`, la méthode `appliquerFiltreEt` qui prend 3 paramètres : une liste de vins et deux pointeurs sur filtre, et qui renvoie une liste de vins par application de la conjonction des deux filtres.

```
listeDeVins appliquerFiltreEt(const listeDeVins &l, const filtre<vin> *f1,
                              const filtre<vin> *f2=nullptr)
{
  listeDeVins l1 = appliquerFiltre(l, f1);
  return (f2 != nullptr) ? appliquerFiltre(l1, f2) : l1;
}
```

.....

► 9. Dans la méthode `main`, ajoutez le code pour extraire de la liste `l`, tous les bordeaux rouges.

```
listeDeVins bordeauxRouge = l.appliquerFiltreEt(l,new filtreBordeaux(),new filtreRouge())
```

.....

► 10. Dans la méthode `main`, ajoutez le code pour extraire de la liste `l`, tous les bordeaux rouges de 2002.

```
bordeauxRouge.appliquerFiltreEt(bordeauxRouge, new filtre2002()));
```

.....