MOOC Init Prog C++
Corrigés semaine 7

Les corrigés proposés correspondent à l'ordre des apprentissages : chaque corrigé correspond à la solution à laquelle vous pourriez aboutir au moyen des connaissances acquises jusqu'à la semaine correspondante.

Exercice 21 : généricité

> Cet exercice correspond à l'exercice n°23 (pages 61 et 228)
> de l'ouvrage *C++ par la pratique* (3e édition, PPUR).

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

int demander_nombre(int a, int b)
{
  /* échange les arguments s'ils n'ont pas été donnés dans *
   * le bon ordre.                                         */
  if (a > b) { int tmp(b); b=a; a=tmp; }

  int res;
  do {
    cout << "Entrez un nombre entier compris entre "
         << a << " et " << b <<" : ";
    cin >> res;
  } while ((res < a) or (res > b));

  return res;
}

int main () {
  double valeur1(3.14159265358);
  double valeur2(1.42857142857);
  double valeur3(-12.344556667);
  double* choix(0);

  switch (demander_nombre(1,3)) {
  case 1: choix = &valeur1; break;
  case 2: choix = &valeur2; break;
  case 3: choix = &valeur3; break;
  }
```

```
  cout << "Vous avez choisi " << *choix << endl;

  return 0;
}
```

Exercice 22 : référence

Cet exercice correspond à l'exercice n°26 (pages 64 et 231) de l'ouvrage *C++ par la pratique* (3e édition, PPUR).

3.1 Références

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

struct Maison {
  string adresse;
};

struct Personne {
  string nom;
  Maison& home;
};

void affiche(const Personne& p) {
  cout << p.nom << " habite " << p.home.adresse << endl;
}

int main()
{
  Maison m1 = { "12 rue du chateau" };
  Personne p1 = { "Pierre", m1 };
  Personne p2 = { "Paul"  , m1 };

  Maison m2 = { "13 rue du chateau" };
  Personne p3 = { "Steve", m2 };
  Personne p4 = { "Sofia", m2 };

  affiche(p1);  affiche(p2);
  affiche(p3);  affiche(p4);

  return 0;
}
```

3.2 Limites des références

Les références ne pouvant pas être modifiées (en tant que telles, i.e. être déplacées), il faut ici les remplacer par des *pointeurs* :

```
#include <iostream>
```

```cpp
using namespace std;

struct Maison {
  string adresse;
};

struct Personne {
  string nom;
  Maison* home;
};

void affiche(const Personne& p) {
  cout << p.nom << " habite " << (*(p.home)).adresse << endl;
  // Note : (*X).Y peut aussi s'écrire X->Y, par exemple ici :
  //      (p.home)->adresse
}

int main()
{
  Maison m1 = { "12 rue du chateau" };
  Personne p1 = { "Pierre", &m1 };
  Personne p2 = { "Paul"  , &m1 };

  Maison m2 = { "13 rue du chateau" };
  Personne p3 = { "Steve", &m2 };
  Personne p4 = { "Sofia", &m2 };

  affiche(p1);  affiche(p2);  affiche(p3);  affiche(p4);

  // déménagement de Pierre (p1)
  p1.home = &m2;
  cout << "maintenant : ";
  affiche(p1);  affiche(p2);  affiche(p3);  affiche(p4);

  return 0;
}
```
