

# Série 7.1

---

## Exercice 1 : Modèle de fonction

1. Écrire un modèle de fonction *exchange* utilisant un paramètre de type  $\langle T \rangle$
2. Réaliser un programme permettant de tester la fonction avec :
  - deux valeurs entières
  - deux valeurs réelles
  - deux objets de type *Time* (série 3.1)\*

\*Question: que doit implémenter la classe *\*Time* au minimum ?

## Exercice 2: Paramètre de type et paramètre expressions

Écrire un modèle de fonction *sum* muni d'un paramètre de type  $\langle T \rangle$  et d'un paramètre expression. La fonction aura deux paramètres:

1. L'adresse d'un tableau dont les éléments sont de type quelconque (float, int, Point);
2. Le nombre d'éléments qu'il faut additionner.

Cette fonction retourne la somme des éléments indiqués par le deuxième paramètre.

On pourra tester cette fonction avec le programme ci-dessous :

```
int main ()
{
    int    intTable[]    = {3, 5, 2, 1};
    float  floatTable[] = {2.5, 3.2, 1.8};
    char   charTable[]  = { 'A', '0', 'i', 'o', 'u' };
    Time   timeTable[]  = { {5, 10}, {3, 22} };

    cout << sum (intTable, 4) << endl;    // type deduction
    cout << sum (floatTable, 3) << endl; // type deduction
    cout << sum (charTable, 5) << endl;  // type deduction
    cout << sum (timeTable, 2) << endl;  // type deduction

    cout << "\nPlease hit ENTER to continue... ";
    cin.get();
    return 0;
}
```