

Cours de C++, 1ère année, HE-Arc

Série 8.2: Exceptions

Exercice 1

Reprendre la classe Vector de la série 3.2 et y ajouter le traitement des exceptions susceptibles d'être levées.

Supposons la fonction main() suivante :

```
int main()
{
    const int SIZE = 2147483647;

    // A) Test de l'allocation dans le constructeur Vector(int, int)
    Vector v1(SIZE);

    // B) Test de l'allocation dans l'opérateur d'assignement
    for() ...
        tabV[i] = v1;
    ...

    // C) Afficher la taille des vecteurs du tableau
    ...

    // D) Test de l'allocation dans le constructeur Vector(int, int)
    Vector v2(SIZE);

    // E) Test de l'allocation dans le constructeur par recopie Vector(int, int)
    Vector v3 = v1;

    // F) Test de l'allocation dans l'opérateur d'affectation
    Vector v5;
    v5 = v1;

    // G) Test de l'allocation dans le constructeur Vector(int, int) avec valeur
    // incorrecte
    Vector v6(-1);

    // H) Test de l'accès hors limites à un vecteur
    v1[-1] = 5;

    // I) Test de l'accès hors limites à un std::vector
    cout << tabV.at(-1).getSize() << endl;

    return 0;
}
```

Des exceptions peuvent se produire dans les situations suivantes :

A) Lors de la création d'un objet de type `Vector(int, int)`

B) Lors de l'affectation d'un vecteur à un autre vecteur

- Construire un `std::vector` contenant 10 vecteurs de type `Vector`, puis copiez le vecteur `v1` dans ce tableau afin de voir apparaitre une exception.
- C) Afficher la taille des vecteurs du tableau

D) Lors de la création d'un objet de type `Vector(int, int)`

E) Lors de la création d'un objet de type `Vector(const Vector&)`

F) Lors de l'affectation d'un vecteur à un autre vecteur

H) Lors d'un accès hors limite à un vecteur

I) Lors d'un accès hors limite à un `std::vector` avec l'opérateur `at()`

Remarque:

- Il ne faut pas mettre l'ensemble du programme dans un `try`
- En cas d'erreur d'allocation d'un Vecteur, il faut s'assurer celui-ci est remis en ordre et le programme `main()` doit continuer.