

BLIN Sébastien
COLLIN Pierre-Henri



École supérieure d'ingénieurs de Rennes

1ère Année
Parcours Informatique

Compte-rendu

TP n°2 : Un vecteur générique

Sous l'encadrement de :
Lamarche Fabrice

1 Une classe de vecteur générique

1.1 Paramètres génériques

Le seul paramètre générique de la classe Vecteur est `T *m_vecteur` ;

1.2 Contraintes associées

Ce type `T` doit redéfinir plusieurs opérateurs notamment l'opérateur `+` qui est utilisé (ou encore l'opérateur `*`).

1.3 Addition de 2 `std::string`

L'addition de 2 `std::string` signifie une concaténation.

2 Des fonctions génériques

2.1 Nouvelles contraintes

Le type `T` doit maintenant redéfinir les opérateurs « et »

2.2 Compilateur et genericité

Une multiplication scalaire de `std::string` ne veut rien dire. Il y a une erreur car l'opérateur `*` n'est pas défini pour le type `std::string`.

Le compilateur compile chaque fonction et chaque classe générique pour chaque type utilisé (et effectue les vérifications à chaque fois).

3 Les types

3.1 Multiplication d'un `int` et d'un `float`

Pour le moment on ne peut faire des multiplication que d'un type (au-dessus de l'opérateur `*` il y a `template<class T>`). Pour régler ce problème il suffit de mettre 2 classes dans le template.

3.2 `a*b` ; `b*a`

Dans notre multiplication, notre résultat est casté dans le type de la première opérande (c'est-à-dire `a*b` sera du type de `a`, `b*a` sera du type `b`)