

# Documentation de validation

Valentin Laclautre, Anthony Dard, Damien Trouche, Martin Gangand,  
Basel Darwish Jzaerly

## Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Bon fonctionnement de decac</b>	<b>2</b>
2.1	Détection des différentes options données . . . . .	2

# 1 Introduction

Cette documentation a pour but de présenter tout les différents tests créés pendant le projet. Pour chaque test, on décrira son fonctionnement, on expliquera son intérêt. On présentera aussi la manière de l'exécuter, ainsi que son importance dans la couverture de test.

## 2 Bon fonctionnement de decac

### 2.1 Détection des différentes options données

Ce premier test a pour but de vérifier que le passage des différentes options données en ligne de commande lors de l'exécution de decac est correct. Pour cela on fait appel à la classe *CompilerOptions* et on regarde la valeur des différents attributs donnant si une option est présente ou non. On vérifie aussi que si l'on donne des options incorrects, une erreur est levée.

## 3 Fichiers \*.deca pour tester la syntaxe contextuelle du langage Deca

Nous trouvons dans test/deca/context/ des fichiers qui respectent la syntaxe du langage Deca et d'autres qui ne la respectent pas. Pour chaque programme valide (qui se trouve dans valid/custom), nous avons en commentaires une description du programme, le résultat attendu et ensuite le programme. La structure est la même pour les fichiers invalides (qui se trouvent dans invalid/custom). Nous voyons en commentaires pourquoi le programme ne compile pas.

---

```
// affect-compatible-type-value.deca
// Description: Affectation d'une valeur sur un type different
//              mais compatibles.
// Resultats:
//   Ligne 12: affichage 1.f.
//   Cela est permis selon la page 75 du poly.
{
    // Doit etre accepte.
    float a = 1;
    print(a);
}
```

---

---

```
// affect-incompatible-type-value.deca
// Description:
//   Affectation et typage sur valeur
// Resultats:
//   Ligne 12: Affectation d'une valeur sur un type incompatibles.
//   Cela est permis selon la page 75 du poly.
// Remarque : la reciproque est accepte.
{
    // Doit etre refuse.
    int a = 1.2f;
}
```

---