

Des principes pour bien programmer

Alexis Nasr (d'après les slides de Arnaud Labourel)



Do One Thing

Une fonction ne doit faire qu'**une seule chose**.

Pour cela, elle ne doit réaliser que des étapes de même niveau d'abstraction.

On décompose la fonction :

Pour faire la cuisine je dois (premier niveau d'abstraction) :

- choisir une recette;
- réunir les ingrédients;
- suivre la recette.

Pour choisir une recette, je dois (deuxième niveau d'abstraction):

- réfléchir à ce que j'ai envie de manger;
- chercher sur marmiton.

Mauvaise approche

```
void cook(){  
    // On choisit la recette  
    Food wantToEat = thinkAboutFood();  
    Recipe recipe = lookOnMarmiton(wantToEat);  
  
    // On réunit les ingrédients  
    openFridge();  
    (for Ingredient ingredient : recipe.getFreshIngredients()){  
        takeInFrige(ingredient);  
    }  
    closeFridge();  
    openCupboard();  
    ...  
  
    // On suit la recette  
    ...  
}
```

Bonne approche

```
void cook(){  
    Recipe recipe = chooseRecipe();  
    gatherIngredients(recipe);  
    followRecipe(recipe);  
}
```

```
void chooseRecipe(){  
    Food wantToEat = thinkAboutFood();  
    Recipe recipe = lookOnMarmiton(wantToEat);  
}
```

...

Un programme est “bien conçu” s’il permet de :

- Absorber les changements avec un minimum d’effort
- Implémenter les nouvelles fonctionnalités sans toucher aux anciennes
- Modifier les fonctionnalités existantes en modifiant localement le code

Objectifs

- Limiter les modules impactés
 - ▶ Simplifier les tests unitaires
 - ▶ Rester conforme à la partie des spécifications qui n’ont pas changé
 - ▶ Faciliter l’intégration
- Gagner du temps