

Programmation: théorie

Romain Deleuze

23 septembre 2019

Table des matières

1	Buts du cours	2
1.1	Réflexion	2
1.2	Gérer votre temps	2
1.2.1	Discussion	2
2	Module 0 : Introduction	3
2.1	Langages de l'informatique	3
2.2	Programmeur et utilisateur	3
2.3	Algorithme	3
2.4	Programme	4
2.4.1	Schéma exécution programme	4
2.4.2	Compiler ou interpréter	4

Chapitre 1

Buts du cours

Lecture 1: Intro

Jeudi 19 Septembre 2019 13 :

1.1 Réflexion

1. Écrire du code propre (clean code)
 - Compréhensible
 - Bien structuré
 - Avec des noms explicites
 - Pouvoir réutiliser son code
2. Réfléchir avant de coder
 - Réfléchir à l'algorithmique
 - Utiliser les instructions les plus adéquates
 - Choisir des structures de données appropriées
3. Comprendre ce que vous lisez/écrivez
 - Pas d'essais/erreurs
 - Pas de rustines
 - Comprendre le "pourquoi" -> déduire correction
4. Pas de recopier le code d'Internet
 - Oublier si incompréhensible
 - Améliorer le

1.2 Gérer votre temps

1. Travailler en dehors des séances d'exercices

1.2.1 Discussion

- Discuter entre vous et poser des questions
- Poser des questions via Discord

Chapitre 2

Module 0 : Introduction

2.1 Languages de l'informatique

1. Languages impératifs (procéduraux)
 - C
 - C++
 - C#
2. Languages fonctionnels
 - F#
 - Haskell
 - Python
3. Languages logiques
 - Prolog
 - Mercury
4. Languages multi-paradigmes
 - Python

2.2 Programmeur et utilisateur

1. Utilisateur : Mot de passe -> accès à un site web
2. Programmeur : Tache -> Quoi faire -> Comment ? -> Algorithme -> Programme

2.3 Algorithme

1. Algorithme exact -> Résultat attendu
2. Algorithme inexact -> Résultat indéfini (fonctionne with edge cases)
3. Exemples

- Recettes de cuisine
- Notice de montage
- 4. Définition algorithmique : Sciences des algorithmes qui incluent l'étude de leur complexité et leur conception

2.4 Programme

Séquence d'instructions exécutables par l'ordinateur -> Intermédiaire humain/-machine

2.4.1 Schéma exécution programme

2.4.2 Compiler ou interpréter

- Compilation : Code source -> Compilation -> Code objet -> Exécution -> Résultat
- Interprétation : Code source -> Interprétation -> Résultat
- Python : Mélange les 2 : Code source -> Compilation -> Code objet -> Interprétation -> Résultat