Série 9: `struct`

Lien avec le MOOC Initiation à la Programmation (en C++)

Exercices semaine6 du MOOC : 2nde partie `struct`

- **Document Tutoriel « définition et utilisation d'une structure Personne »**

- **Document Exercices semaine 6 du MOOC (partie struct)**

  - **Exercice 19 : nombres complexes (niveau 1)**
    - passage de structure en paramètre à une fonction

  - **Exercice 20 : QCM (niveau 2)**
    - structure et vector

- **Document Exercices additionnels semaine 6 du MOOC (partie struct)**

  - **Exercice 17 : fractions (niveau 2)**
    - Illustration du principe de séparation des fonctionnalités : une seule fonction est responsable de la simplification d'une fraction ; toutes les autres fonctions l'utilisent pour simplifier le résultat de l'opération dont elles sont responsables.

Complément Projet (5)

l'écriture et les test des premières fonctions peut commencer en testant d'abord une seule fonction, puis une fois validée, on peut passer au test de la fonction suivante etc… Dans ce contexte, le fait de travailler avec des vector peut conduire à un arrêt brutal du programme avec « segmentation fault » ou « bus error » « core dump »… Il faut alors identifier la ligne de code fautive. La localisation en faisant afficher des messages avec cout est populaire mais pas toujours efficace.

Le débugger ddd peut nous aider pour trouver les bugs pendant l'exécution:

- Pour utiliser un programme de déverminage, il faut ajouter l'option **`-g`**
  - **`g++ -Wall –std=c++11 -g -o monprogramme monprogramme.cc`**

- Lancer le dévermineur dans un terminal : **`ddd monprogramme`**
- Démarrer **`monprogramme`** dans **`ddd`** avec run ou run *arguments* en cas de redirection
- Après un crash, choisir Backtrace dans le menu **status** pour savoir l'état de la Pile
  - On y voit la succession d'appels de fonctions à partir de main()

- Pour suspendre l'exécution du programme à des endroits précis utiliser le bouton « **Break** »
- Pour exécuter pas à pas : **`next`** ou **`step`**
- Pour regarder le contenu d'une variable :
  - mettre la souris dessus
  - écrire dans la fenêtre ddd la commande **`print nom_variable`**
  - écrire dans la fenêtre ddd la commande **`display nom_variable`**
    - La valeur de la variable est alors affichée à chaque pas de programme.