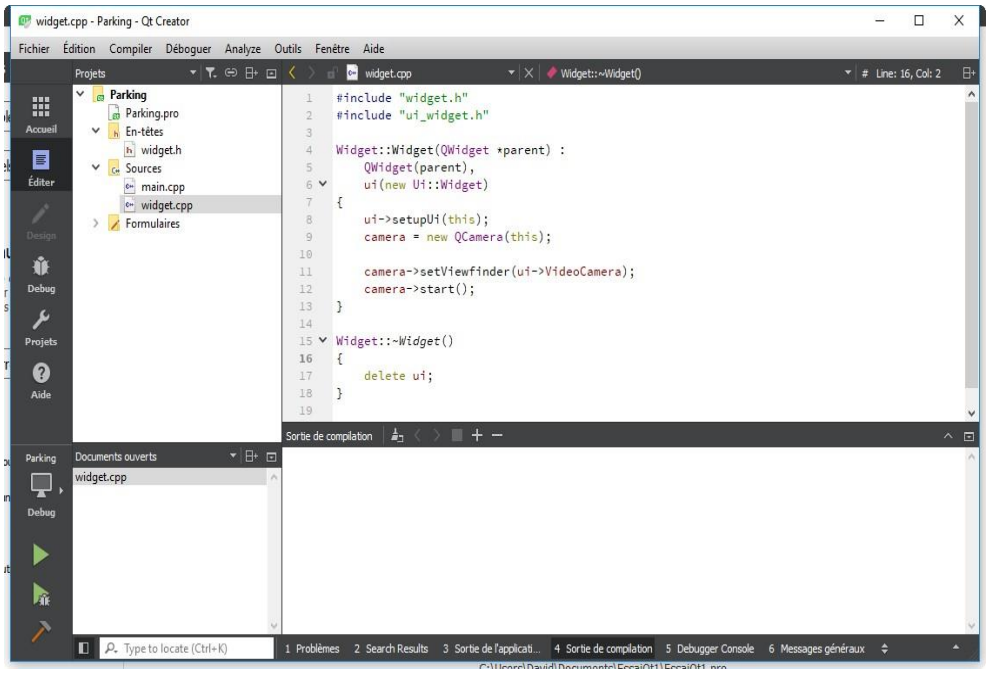
TP N°3



25/09/2018

Développement en C/C++ sous QtCreator

L'objectif de ce TP est de vous familiariser avec l'environnement de développement intégré (EDI) QtCreator par le biais d'exercices.

TP n°3

DEVELOPPEMENT EN C/C++ SOUS QTCREATOR

1. Introduction

QtCreator est un environnement de développement intégré : il regroupe un éditeur de textes avec coloration syntaxique mais aussi une analyse syntaxique à la volée ; un ensemble de développement (compilateur, éditeur de liens, debugger) que l'on appelle *toolchain*.

Plus généralement QtCreator fait partie d'un ensemble logiciel Qt (prononcez « quiuutie » à l'anglaise) permettant de développer des applications (logiciels) sous Windows, Linux, Androïd, Ios. En fait, Qt est une librairie logicielle qui vient s'ajouter à la librairie standard du C/C++. On dit que c'est une surcouche logicielle. Elle simplifie le développement en mettant à la disposition du développeur des fonctions, des classes directement exploitables pour réaliser des tâches qui demandaient auparavant des milliers de lignes de code.

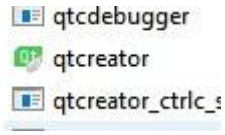
Remarque : l'environnement KDE qu'on trouve dans certaines distributions Linux est basé sur Qt

2. Prise en main de l'EDI

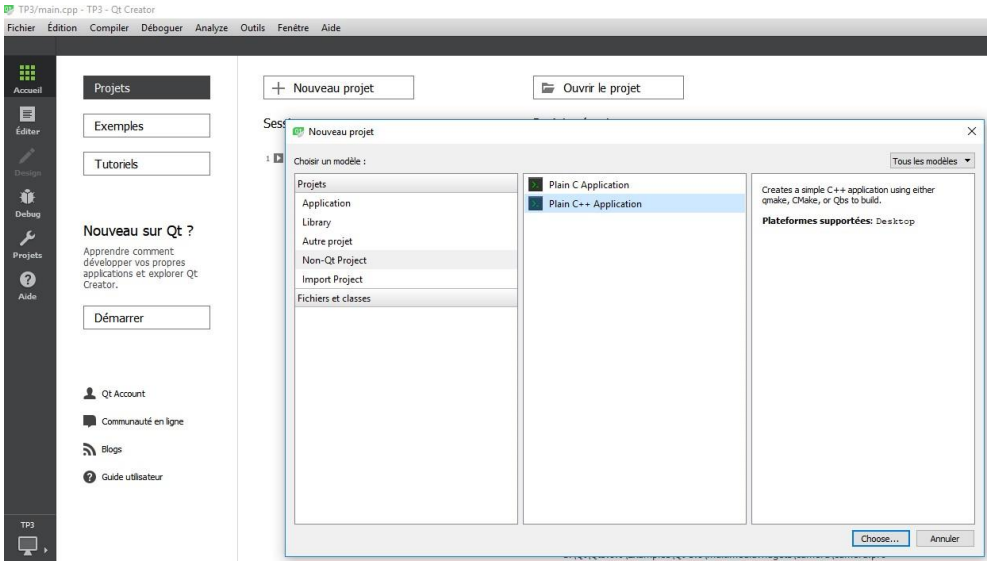
L'exécutable de QtCreator sous Windows se situe dans le répertoire d'installation de Qt : généralement *C:\Qt\Qt5.8.0\Tools\QtCreator\bin*

Qt5.8.0 est la version de Qt installée : il est possible que vous ayez une version différente !

Double cliquer sur l'exécutable *QtCreator*

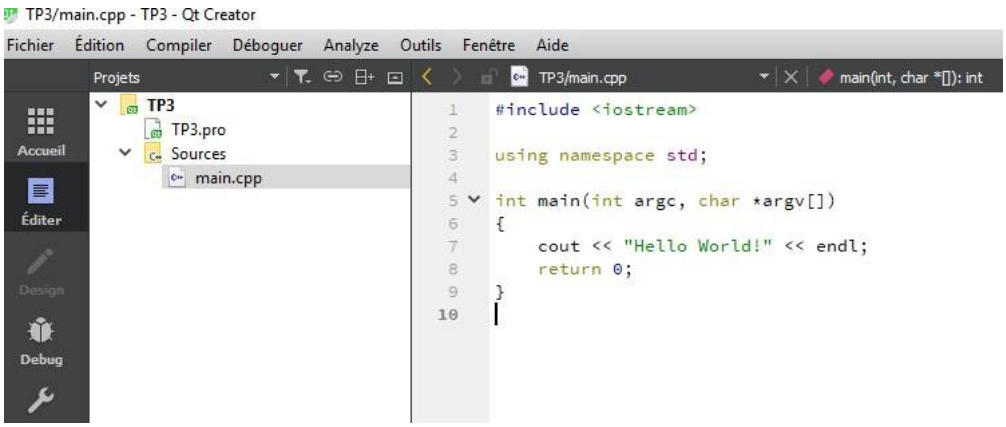


Dans la fenêtre principale de QtCreator, choisir un nouveau projet et sélectionner un « None-Qt Project / Plain C++ Application »



Vous lui donnerez un nom de projet (par exemple TP3_1) ; cliquer sur suivant jusqu'à « Terminer »

Vous devriez obtenir quelque chose comme :



Vous reconnaissez la fonction « main » ainsi que la chaîne de caractères « Hello World ! »

On va produire l'exécutable correspondant à ce fichier source : Vous devez cliquer sur le triangle vert en bas à gauche de la fenêtre :

Le triangle vert permet de compiler, linker et exécuter le programme

Le triangle vert avec une bestiole permet de compiler, linker et exécuter le programme en mode débugger (pas à pas)

Le marteau permet de compiler et linker le programme

Explications sur l'écran/Debug : on peut choisir de produire l'exécutable en mode Debug (exécutable muni des informations de débogage : table des symboles, code source …)

On peut choisir de générer l'exécutable sans débogage mais prêt à être utilisé en dehors de QtCreator : on dit qu'on génère une release. Généralement pendant le temps de développement et tests, on travaille en mode Debug ; une fois que l'application est prête, on la génère en Release afin qu'elle puisse être exécutée sur d'autres machines.

Les exécutables produits à l'issue de la compilation se trouvent dans des répertoires dont le nom commence par *build* … Par exemple chez moi, l'exécutable Release de TP3 se trouve dans *G:\lycee\SN1\2017\C\TP\build-TP3-Desktop_Qt_5_8_0_MinGW_32bit-Release\release*

Sur votre ordinateur, le répertoire *build-*… se trouve éventuellement ailleurs !

3. Premiers programmes dans QtCreator

Saisir le code suivant :

```
#include <iostream>

using namespace std;

int main(int argc, char *argv[])
{
    int    age;    //Réserve une région de la mémoire destinée à contenir un entier et lui attribue le nom age
    string    nom; //Réserve une  région de la mémoire destinée à contenir une chaîne de caractères et lui attribue le nom nom

    cout<<"Donnez votre age : " << endl;  //Affiche un message d'invite demandant à l'utilisateur d'exécuter une action
    cin>>age; //Lit la valeur saisie et la place dans age
    cout<<"Donnez votre nom : " << endl;
    cin>>nom; //Lit les caractères et les place dans nom

    cout<<nom<<" , vous avez "<<age<<" ans"<<endl;

    return 0;
}
```

Compiler et produire l'exécutable.

Constater la différence entre l'instruction *cout* et *cin*.

L'opération qui consiste à faire *cin*>>*age* est une affectation : on prend la valeur tapée au clavier et on la met en mémoire à l'emplacement de « age ».

Ajouter le code suivant au programme :

```
age = age+10;
cout << nom << ", dans 10 ans, vous aurez " << age << " ans" << endl;
```

De sorte que le résultat de l'exécution soit :

```
Donnez votre age :
47
Donnez votre nom :
david
david , vous avez 47 ans
david, dans 10 ans, vous aurez 57 ans
```

Affectation

Dans le code, on effectue *age* = *age* + *10*

Il s'agit d'une affectation.

ATTENTION : IL NE S'AGIT PAS D'UNE EGALITE !

En mathématiques, on ne peut pas écrire x=x+10 quel que soit x

En informatique, on peut puisqu'il s'agit d'évaluer (de calculer) le membre de droite du signe =

Ici, on prend la valeur qui se trouve dans *age*, on lui ajoute 10 puis le résultat est mis dans le membre de gauche donc, ici, *age*

On écrase l'ancienne valeur de *age* par la nouvelle

4. Exercices

4.1 Exercice 1

En vous inspirant du programme précédent, créer un nouveau projet permettant d'avoir le résultat suivant :

```
Donnez votre age :
47
Donnez votre nom :
david
david , vous avez 47 ans
david, dans 10 ans, vous aurez 57 ans
david, vous etes ne en 1970
```

Vous créerez une variable de type *int* qui s'appellera *anneenaissance*

4.2 Exercice 2

On désire désormais tester l'âge saisi par l'utilisateur et vérifier que la saisie n'est pas farfelue. On va donc vérifier que l'utilisateur ne saisit pas un âge négatif.

L'algorithme est très simple :

Pseudo code

```
Lire age
Si age < 0
Alors
    Afficher « age négatif !! »
Sinon
    Afficher « vous avez : » age « ans »
finsi
```

C/C++

```
cin>>age;

if (age<0)
{
    cout << "age négatif"<<endl;
}
else
{
    cout << "Vous avez " <<age<<" ans"<<endl;
}
```

Modifier le programme en conséquence.

Désormais, on voudrait produire un affichage différent en fonction de la tranche d'âge :

De 0 à 18 ans : « Vous êtes élève »

De 18 à 30 ans : « Vous êtes étudiant »

De 30 à 60 ans : « Vous êtes un actif »

De 60 à 75 ans : « Vous êtes un retraité »

Au-delà de 75 ans : « Vous êtes un sénior »

Proposer une solution en modifiant le programme précédent.

4.3 Exercice 3

L'une des premières modifications consistait à tester la valeur de « age » saisie par l'utilisateur. Le problème est que si l'utilisateur saisit un âge négatif, le programme s'arrête.

On voudrait redemander à l'utilisateur de saisir de nouveau un âge et redemander tant qu'il ne saisit pas une valeur positive.

Voici l'algorithme :

Pseudo code

```
Lire age
Tant que age <= 0
    Afficher « veuillez saisir un age positif »
    Lire age
Fin tant que
```

C/C++

```
cin>>age; //Lit la valeur saisie et la place dans age

while (age<=0)
{
    cout << "Veuillez saisir un age positif"<<endl;
    cin>>age;
}
```

Modifier le code en conséquence

4.4 Exercice 4

Réaliser un programme permettant de saisir deux noms et deux âges. Le programme affichera le nom et l'âge du plus âgé.

4.5 Exercice 5

Réaliser un programme permettant la saisie des noms et âges de 10 personnes. Le programme affichera alors la moyenne d'âge de ces 10 personnes.

Aide : pour effectuer plusieurs fois la même chose, on utilise le concept de boucles. On a vu précédemment la boucle *while* : elle est utilisée quand on ne sait pas combien de tours de boucle on effectuera.

Ici, on sait qu'on doit effectuer 10 tours de boucle. On utilise alors la boucle *for*

Pseudo code

```
Moyenne est un réel
Somme est un entier
i est un entier
Somme <- 0
Pour i allant de 0 à 9
    Lire age
    Somme <- Somme + age
    Lire nom
Fin Pour
Moyenne <- Somme / 10
Afficher « Moyenne d'age = » Moyenne
```

C/C++

```
float Moyenne;
int Somme;
int i;
Somme = 0;
for(i=0;i<=9;i++)
{
    A COMPLETER
}
Moyenne = Somme/10.;
cout << "Moyenne d'age = "<<Moyenne << " ans"<<endl;
```