

# Niveau 4

TP  
S4. Développement logiciel  
S4.1. Principes de base  
S4.4. Programmation procédurale

## Prise en main de la chaîne de développement

### TABLE DES MATIÈRES

1 - Présentation.....	2
1.1 - <b>Qt Creator</b> , l'EDI.....	2
1.1.1 - Sélecteur de modes.....	3
1.1.2 - Compilations.....	3
2 - Travail à réaliser.....	4
2.1 - Exercice 1.....	4
2.2 - Exercice 2.....	5
2.3 - Exercice 3.....	5
2.4 - Exercice 4.....	6
2.5 - Exercice 5.....	6
2.6 - Exercice 6.....	6
2.7 - Exercice 7.....	7
2.8 - Exercice 8.....	7
2.9 - Exercice 9.....	7
2.10 - Exercice(s) 10.....	7
3 - Bonus !.....	8



# 1 - PRÉSENTATION

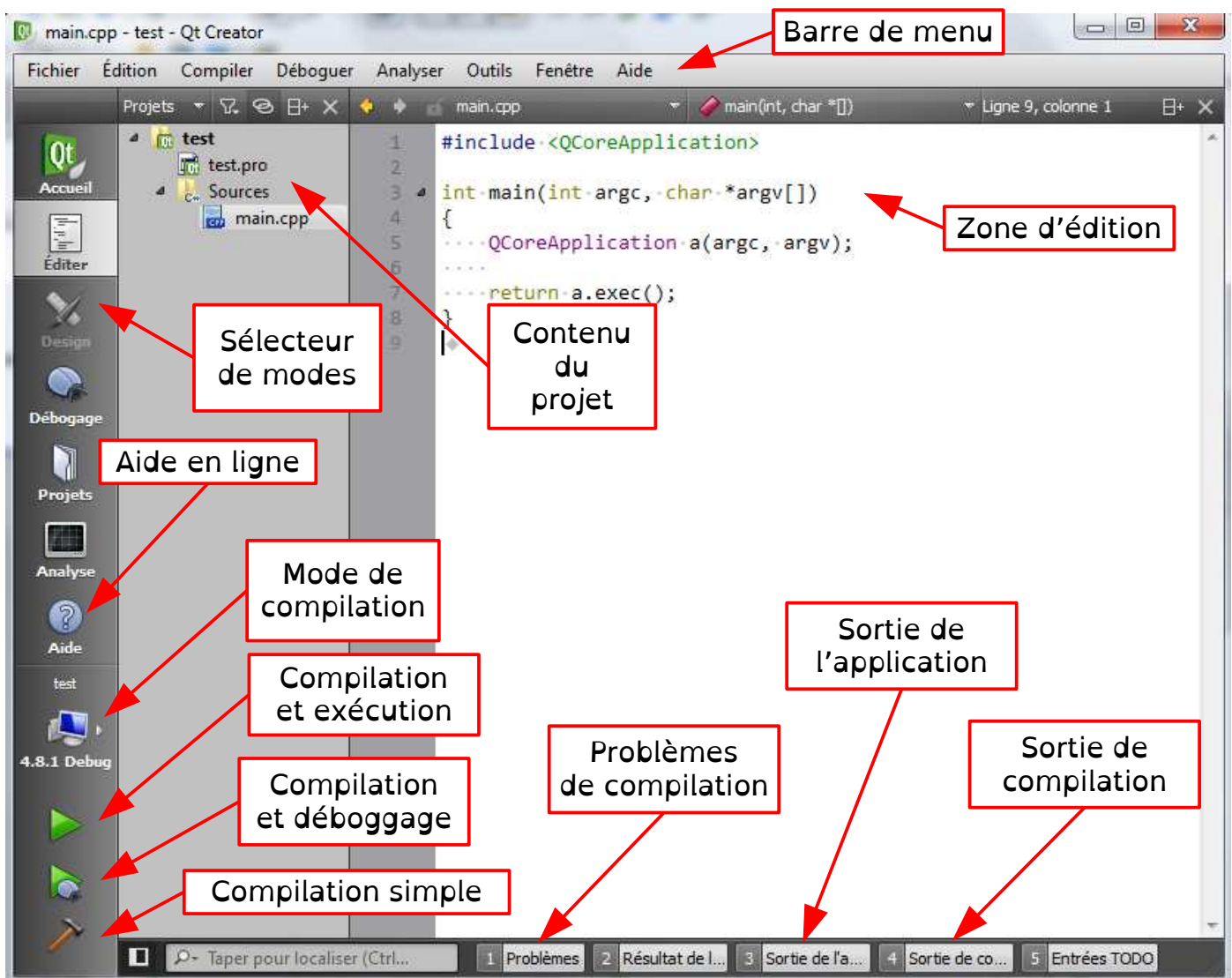
Ce TP a pour but de présenter les fonctionnalités de base de l'E.D.I. (Environnement de Développement Intégré) **Qt Creator**, utilisé dans la section **SN-IR**.

Cet outil permet la programmation de programmes C et C++, basé sur le compilateur MinGW, les bibliothèques STL (**Standard Template Librarie**) et Qt.

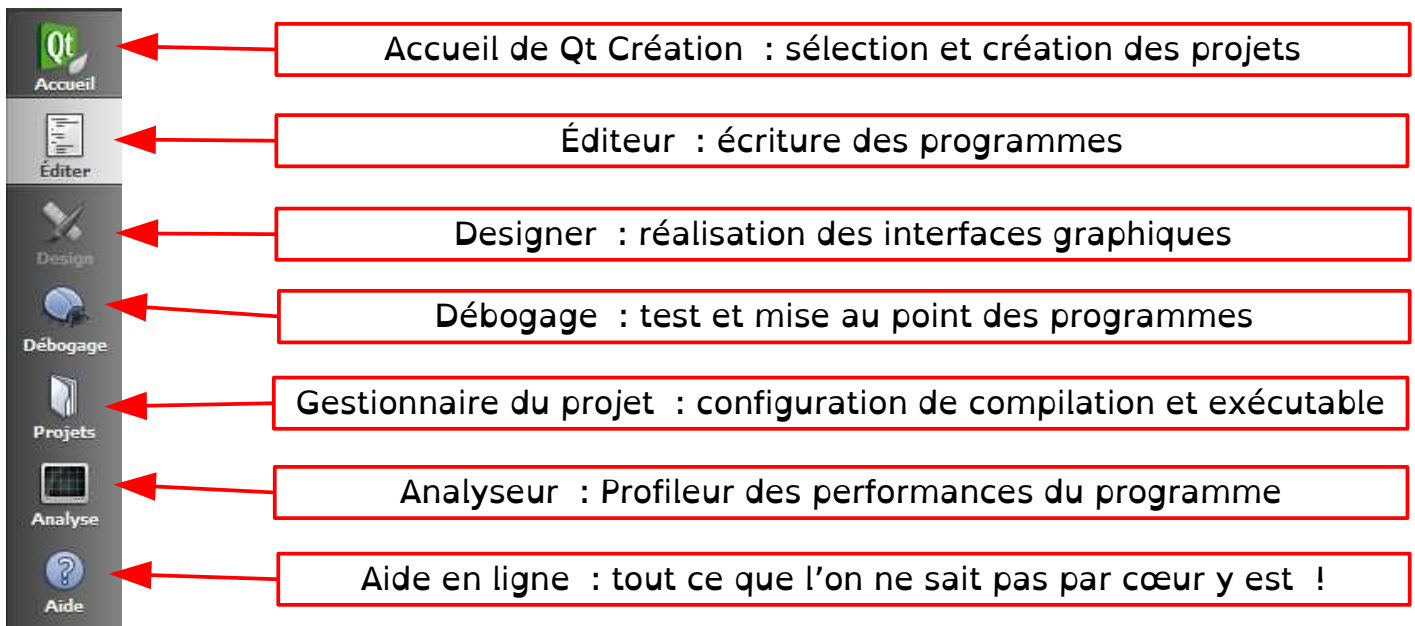
Chacun de ces outils sont parfaitement portables et utilisables indifféremment sous Windows™ et Linux.

## 1.1 - QT CREATOR, L'EDI

L'interface principale est la suivante :



## 1.1.1 - SÉLECTEUR DE MODES



Lors de la création d'un nouveau **projet** de développement, le programmeur doit choisir le type d'application qu'il souhaite réaliser. Parmi tous ceux proposés, les deux plus fréquents sont les projets « **QT Console Application** » et « **Application QT avec Widgets** ».

Notez que le bouton « **Design** » n'est accessible que si le projet en cours est de type « **Application QT avec Widgets** », mode dans lequel les interfaces graphiques sont possibles.

Un projet « **QT Console Application** » ne permet pas d'application graphique et s'exécute dans la console de commande du système d'exploitation. Il s'agit du mode de projet par défaut pour ce TP !

## 1.1.2 - COMPILATIONS

Il existe plusieurs boutons permettant la compilation des programmes :



### **Compilation et exécution**

Ce bouton commande la compilation de tous les fichiers modifiés depuis la dernière fois et, dans le cas où elle se déroule parfaitement, lance l'exécution du programme.



### **Compilation et débogage**

Ce bouton commande la compilation de tous les fichiers modifiés depuis la dernière fois et, dans le cas où elle se déroule parfaitement, lance l'outil de débogage pour la mise au point du programme.



### **Compiler le projet**

Ce bouton compile seulement les fichiers du projet.

**Attention** : Parfois, cette compilation ne tient pas compte de certaines modifications.

Préférez alors l'option **Nettoyer le projet...** puis l'intitulé **Tout recompiler** du menu **Compiler**

## 2 - TRAVAIL À RÉALISER

- 1) Créez un nouveau projet de développement de type « **QT Console Application** » que vous nommerez **Mon\_Premier\_Programme**. Quels sont les fichiers et directories automatiquement générés par l'EDI ?

### 2.1 - EXERCICE 1

- 2) Tapez le programme suivant à l'intérieur de la fonction principale **main(...)** (en remplaçant son contenu !), dans un nouveau projet, enregistrez-le, compilez et exécutez. Que fait-il ?

```
int a   (0) ;
int b   (0) ;
int x   (0) ;
int y   (0) ;

a = 4 ;
b = 7 ;

cout << "A vous de jouer" << endl ;

cin >> x ;
cin >> y ;

if((x==a) && (y==b))
{
    cout << "BOOM ! Coulé !" << endl ;
}
else
{
```

```

    if((x==a) || (y==b))
    {
        cout << "SPLASH ! Éclaboussé !" << endl ;
    }
    else
    {
        cout << "PLOUF ! Dans l'eau !" << endl ;
    }
}

```

- 3) Quels sont les nouveaux dossiers et fichiers qui ont été créés éventuellement dans le directory du projet ?

- 4) Modifiez le programme afin qu'il affiche « À toi de jouer » et non « À vous de jouer ».
- 5) Puis, pour que le bateau soit sur la case de coordonnées (6 ; 9).

## 2.2 - EXERCICE 2

En général, à la bataille navale, un bateau n'est « SPLASH ! Éclaboussé » que si la case visée est immédiatement voisine de celle du bateau.

- 6) Modifier le premier programme de ce chapitre pour tenir compte de cette règle. On pourra traiter le cas où les cases diagonalement adjacentes au bateau sont éclaboussées et le cas où elles ne le sont pas.

## 2.3 - EXERCICE 3

Le programme suivant permet de calculer le prix toutes taxes comprises d'un article, connaissant son prix hors taxes, dans le cas où le taux de TVA est de 19,6% :

```

cout << Quel est le prix hors taxes ? << endl ;
cin >> PrixHorsTaxe ;
PrixTTC = PrixHorsTaxe + " PrixHorsTaxe " * 19.6 / 100 ;
cout << Le prix TTC est << PrixTTC << " endl " ;

```

- 7) Tapez ce programme, dans un nouveau projet, en apportant les modifications nécessaires pour qu'il n'y ait pas d'erreur à la compilation.
- 8) Adaptez-le pour permettre à l'utilisateur de choisir le taux de TVA, qui est de 20 % actuellement.
- 9) Modifiez finalement ce programme pour que les quatre taux de TVA actuellement en vigueur dans nos magasins soient pris en compte ! Pour info :

- taux normal : 20 % ;
- taux intermédiaire : 10 % ;
- taux réduit : 5,5 % ;
- taux particulier : 2,1 %.

## 2.4 - EXERCICE 4

```
cin >> a ;
cin << b ;
cin >> c ;
cin << d ;

if((b == 0) || (d == 0))
{
    cout >> "Denominateur nul interdit" << endl ;
}
else
{
    cout >> a/b << endl ;
    cout >> c/d << endl ;
}
```

10) Que fait cet extrait de programme ?

11) Tapez-le en apportant les modifications nécessaires pour qu'il n'y ait pas d'erreur à la compilation.

12) Comment devrait-on renommer ses variables ?

## 2.5 - EXERCICE 5

13) Écrivez un programme qui, étant donné une équation du second degré, détermine le nombre de solutions réelles et leurs valeurs éventuelles. (Ça me rappelle un algo déjà réalisé en TD... non ?)

## 2.6 - EXERCICE 6

VARIABLES :

n : Entier  
s : Entier  
i : Entier  
u : Entier

DEBUT :

LIRE n  
s <- 1  
u <- 1  
POUR i ALLANT DE 1 A n-1 FAIRE

```
u <- u+2
s <- s+u
FIN POUR
AFFICHER s
FIN
```

- 14) Programmez cet algorithme et exécutez-le.
- 15) Proposez un programme simplifié, permettant la même finalité.

## 2.7 - EXERCICE 7

- 16) Écrivez un programme qui affiche textuellement un calendrier pour une année entière sous la forme :

```
Lundi 1 janvier
Mardi 2 janvier...
```

On commencera toujours au **lundi 1<sup>er</sup> janvier** et non à la réalité actuelle de l'année !

## 2.8 - EXERCICE 8

- 17) Écrivez un programme permettant de jouer au « nombre mystère » : Cf. l'algorithme réalisé pour le jeu révolutionnaire d'Yvon Enbavé, lors d'un précédent TP.

## 2.9 - EXERCICE 9

- 18) Écrivez un programme qui recueille au clavier les températures de 7 jours successifs au fil de l'eau et calcule la température moyenne de la semaine.
- 19) Modifiez le programme précédent pour que l'utilisateur puisse préciser le nombre de jours avant de donner les températures.

## 2.10 - EXERCICE(S) 10

- 20) Reprenez les algorithmes du premier TP d'algorithmie (4.1 - Petits algos rigolos) et codez-les en C++.

Attention : les structures réduites utilisées par AlgoBox vous ont obligé à n'utiliser que les SI, TANT QUE et les POUR, reniant les REPETER JUSQU'À et les SELON ! Utilisez les vraies bonnes structures en C++ : Le C++ vous le permet ! Utilisez les bonnes structures algorithmiques selon le tableau de détermination donné dans le cours !

## 3 - BONUS !

21) Écrivez un programme inédit, ne ressemblant à aucune des problématiques déjà étudiées, ni en TD, ni en TP, en respectant les consignes suivantes :

- les variables sont nommées conformément à leur utilisation invariante ;
- les structures algorithmiques sont adaptées au problème ;
- il faut utiliser au moins une structure itérative et une structure conditionnelle ;
- il faut au moins une acquisition clavier et un affichage.