

Analyse UML et Codage C++

Projet Horloge sous Linux

~ Séquence n°2 ~

Version 3.0 – janvier 2019

Objectif

- Codage C++
 - Classe
 - Instance
 - Constructeur
 - paramètres par défaut
 - Surcharge d'opération
 - Librairie ncurses

Conditions de réalisation

Ressources utilisées :

Matériel

Un PC sous linux

Logiciel

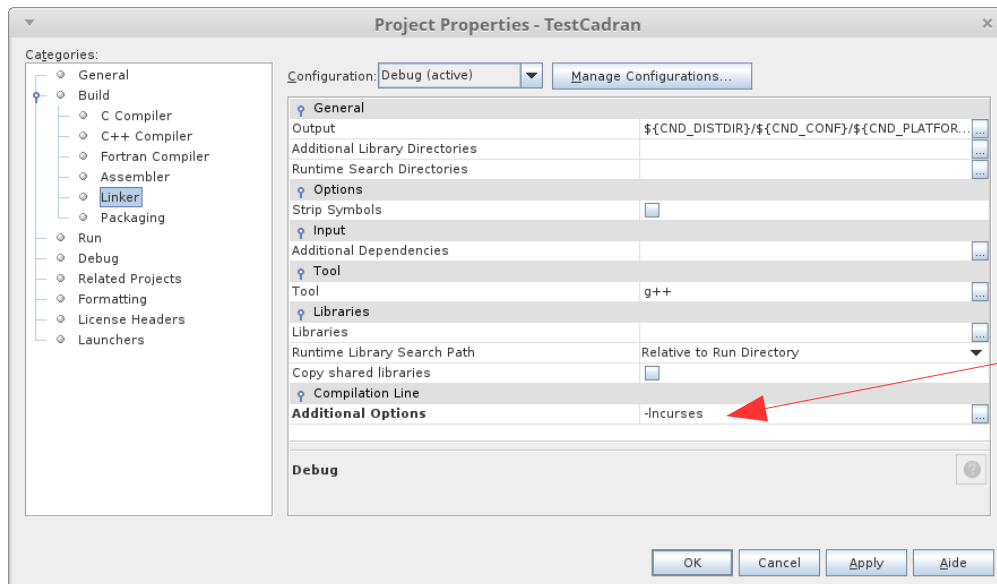
NetBeans

1. CONSTRUCTION DU PROJET SOUS NETBEANS

1. Réalisez un projet nommé **TestCadran** de type application C/C++ sous NetBeans.

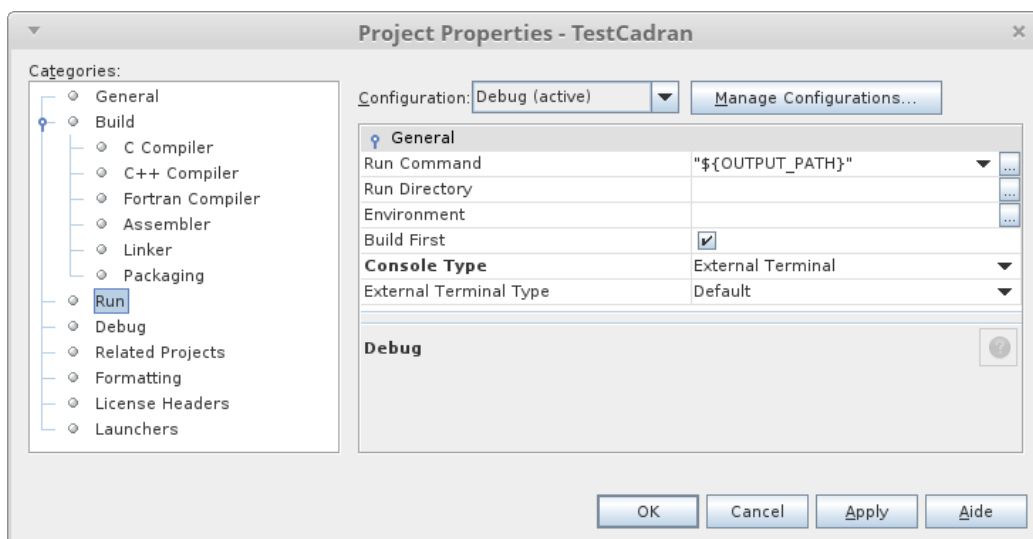
Chaque classe fera l'objet de fichiers séparés (**h et Cpp**) de même pour le programme principal. Ajouter les fichiers **cadran.h** et **cadran.cpp** à votre projet.

2. Modifiez les propriétés de votre projet pour utiliser la librairie **ncurses**.



-Incurses dans
Additional Options
de l'éditeur de lien

3. Toujours dans les propriétés du projet, configurez l'exécution pour utiliser un terminal externe. Au lancement du programme, une nouvelle fenêtre console va apparaître.



4. En utilisant la fonction **sprintf ()** pour la mise en forme du texte, terminer le codage de la méthode :

void Cadran::Afficher(const int valeur, const int position)

2. UTILISATION DE LA CLASSE CADRAN

5. Dans le programme principal, déclarez une instance de la classe **Cadran** nommée **leCadran**, cette instance utilisera les paramètres par défaut du constructeur.

Utilisez les méthodes **void Cadran::Afficher()** pour faire défiler un compteur variant de 0 à 10 en position 3. Pour ralentir l'affichage on propose d'utiliser la fonction `sleep` de la librairie `<unistd.h>`. Lorsque l'affichage est terminé, inscrire le mot "Fin" en position 1.

Peut-être sera-t-il nécessaire d'ajouter la portion de code suivante à la fin de votre programme principal pour voir le mot FIN dans la cadran :

```
char touche;
do
{
    touche = getchar();
} while(touche != ' '); // touche ESPACE pour quitter
```

6. Dans ce même programme principal, ajoutez une deuxième instance de la classe **Cadran** aux coordonnées 10,15 sur la console. Réalisez dans la même boucle un décompteur de 10 à 0 en position 5 puis affichage du mot "Fin" en position 1. Utilisez la méthode **void Cadran::Effacer()** si besoin pour obtenir un meilleur résultat.
7. Modifier les méthodes **Afficher** pour que par défaut le texte ou la valeur entière se place à la position 1. Testez les modifications que vous avez effectués.

Est-il possible de positionner un cadran en 0,0 avec 1 ligne et 10 caractères au lieu de 7 en agissant uniquement sur le paramètre **_largeur** du constructeur de la classe **Cadran** ?
8. D'écrire les différentes fonctions de la librairie `<ncurses>` utiliser dans le constructeur, le destructeur et dans la méthode **Afficher** de la classe **Cadran**.
9. Terminer le commentaire des méthodes de la classe **Cadran**.

```
/**
 * @file Cadran.cpp
 * @author philippe CRUCHET
 * @date 31 décembre 2018, 16:09
 */

#include "Cadran.h"
/**
 * Rôle : Constructeur de la classe, Initialise le cadran
 * et trace le contour.
 * @param _posX position en X coin haut-gauche du cadran
 * @param _posY position en Y coin haut-gauche du cadran
 * @param _hauteur nombre de ligne d'affichage
 * @param _largeur nombre de caractères par ligne
 */
Cadran::Cadran(const int _posX, const int _posY, const int _hauteur, const int _largeur) :
posX(_posX),
posY(_posY),
hauteur(_hauteur),
largeur(_largeur)
{
    initscr();
    keypad(stdscr, TRUE);
    noecho();
    mvaddch(posY, posX, '+');
    mvaddch(posY, posX + largeur + 1, '+');
    mvaddch(posY + hauteur + 1, posX, '+');
    mvaddch(posY + hauteur + 1, posX + largeur + 1, '+');
    mvhline(posY, posX + 1, '-', largeur);
    mvhline(posY + hauteur + 1, posX + 1, '-', largeur);
    mvvline(posY + 1, posX, '|', hauteur);
    mvvline(posY + 1, posX + largeur + 1, '|', hauteur);
    curs_set(0);
    refresh();
}

/**
 * Rôle : Destructeur de la classe
 */
Cadran::~~Cadran()
{
    endwin();
}

/**
 * Rôle : Affiche un texte à la position donnée
 * @param texte Chaîne de caractères à afficher
 * @param position position du 1er caractère, 0 = 1ère position
 */
void Cadran::Afficher(const char *texte, const int position)
{
    mvprintw(posY + 1, posX + position + 1, texte);
    refresh();
}
```

```

/**
 * Rôle : Affiche un entier sur 2 chiffres à la position données
 * @param valeur  entier à afficher
 * @param position position de la dizaine, 0 = 1ère position
 */
void Cadran::Afficher(const int valeur, const int position)
{
    char *texte = new char[largeur + 1];
    if (texte != NULL)
    {
        // à compléter
        Afficher(texte, position);
        delete texte;
    }
}

/**
 * Rôle : Efface la zone d'affichage de l'écran
 */
void Cadran::Effacer()
{
    mvhline(posY + 1, posX + 1, ' ', largeur);
    refresh();
}

```

```

/**
 * @file: Cadran.h
 * @author: philippe
 *
 * @date: 31 décembre 2018, 16:09
 */
#ifndef CADRAN_H
#define CADRAN_H

#include <ncurses.h>

class Cadran {
public:
    Cadran(const int _posX=1,const int _posY=1,const int _hauteur=1,const int _largeur=7);
    virtual ~Cadran();
    void Afficher(const char *texte, const int position);
    void Afficher(const int valeur,const int position);
    void Effacer();

private:
    int posX;
    int posY;
    int hauteur;
    int largeur;

};

#endif /* CADRAN_H */

```