

Projet Horloge

~ TD4 ~

Analyse UML ~ Codage C++ ~ Heritage

Version 4.0 – septembre 2020

Objectif

- Codage C++
 - Héritage
 - Accès protégé
 - Résolution de portée
- UML
 - Relations entre classes : Héritage, Composition
 - Diagramme de séquence

Conditions de réalisation

Ressources utilisées :

Matériel

Un ordinateur sous Linux

Logiciel

QtCreator
Projet sans la librairie Qt en
mode console
Modelio

1. SUITE DE L'EXTRAIT DU DOSSIER D'ANALYSE

Description des classes

Classe : 'Reveil'

Description :

Cet objet gère l'heure courante et dispose d'opérations pour la mise à l'heure.

Attributs : [Privé]

heuresAlarme <i>integer</i>	Heure de l'alarme, valeur comprise entre 00 et 11 ou entre 00 et 23 suivant la résolution.
minutesAlarme <i>integer</i>	Minute de l'alarme, valeur comprise entre 00 et 59
alarme <i>booléen</i>	Indique si l'alarme doit être surveillée
enAlarme <i>booléen</i>	Indique que l'alarme a été déclenchée.

Opérations : [Public]

Reveil (<i>in short _nbModes, in short _resolution</i>) par défaut <i>_resolution = 24</i> et <i>_nbModes = 6</i>	
	Constructeur de la classe, initialise les attributs heuresAlarme et minutesAlarme à 0, l'attribut alarme est mis à FAUX. Il crée également l'instance de la sonnerie.
~Reveil ()	
	Destructeur de la classe.
void AvancerHeuresAlarme ()	
	Passe à l'heure de l'alarme suivante en tenant compte de la résolution .
void AvancerMinutesAlarme ()	
	Passe à la minute de l'alarme suivante.
void ReculerHeuresAlarme ()	
	Revient à l'heure Alarme précédente.
void ReculerMinutesAlarme ()	
	Revient à la minute alarme précédente.
TOUCHE_CLAVIER Controler (<i>in TOUCHES_CLAVIER _numTouche</i>) par défaut <i>_numTouche = 0</i>	
	Assure le fonctionnement du réveil pour le mode dans lequel elle se trouve. Elle reçoit le numéro de la touche enfoncée, agit en accord avec le diagramme états-transitions de la classe Reveil. Assure les affichages correspondant au mode et retourne numéro de la touche enfoncée.

Classe : 'Sonnerie'**Description :**

Cet objet gère les actions de l'utilisateur sur le clavier

Opérations : [Public]

Sonnerie ()	
	Constructeur de la classe.
~Sonnerie ()	
	Destructeur de la classe.
void FaireSonner ()	
	Fait retentir l'alarme
void ArrêterAlarme ()	
	Stop le bruit de l'alarme

2. MISE A JOUR DE L'ANALYSE UML

1. Sous Modelio Importer le projet Reveil fournit en ressources
2. Modifiez le diagramme des cas d'utilisation en ajoutant les cas « **Surveiller Alarme** » et « **Regler Alarme** ».
3. Complétez le diagramme de classe en ajoutant la classe **Reveil** et la classe **Sonnerie** en respectant les descriptions données précédemment. Ajoutez les relations entre les différentes classes sachant qu'un réveil est une sorte d'horloge et qu'il est composé d'une sonnerie.
4. Réalisez le diagramme états-transitions de la classe **Reveil**. En plus du mode AUCUN_REGLAGE, on trouve les états REGLAGE_HEURES_ALARME, REGLAGE_MINUTES_ALARME et ACTIVATION_ALARME.
5. Réalisez le diagramme de séquence du cas d'utilisation « **Surveiller Alarme** » à partir de la description suivante :

Si le réveil est dans le mode AUCUN_REGLAGE, si l'alarme est active lorsque l'attribut **heures** est égal à **heuresAlarme** et l'attribut **minutes** est égale à **minutesAlarme**, la sonnerie doit être activée.

3. CODAGE C++ DE L'HÉRITAGE

6. Créez un nouveau Projet nommé **ProjetReveil** de type application C++ sous **QtCreator** dans le dossier **ProjetHorloge**, comme chaque fois le suivi de version est réalisé par Git. Copiez les fichiers des classes **Cadran**, **Clavier** et **Horloge** dans ce projet.
7. Créez la classe **Reveil** sans omettre la relation d'héritage avec la classe **Horloge**. Réalisez le constructeur et le destructeur de cette classe. Dans votre programme principal, instanciez la classe **Reveil**, le cadran de l'horloge doit s'afficher lors de l'exécution.
8. Créez la classe **Sonnerie**, sans omettre la relation avec la classe **Reveil**.
9. En vous inspirant du travail réalisé sur la classe **Horloge** et des descriptions fournies, codez les différentes méthodes de la classe **Reveil** sauf la méthode **Controler**.

Toutes les données membres dont vous avez besoin dans les différentes méthodes sont-elles accessibles dans la classe **Reveil** ? Faites les opérations nécessaires et modifiez votre analyse UML sous modelio.

10. Réalisez une constante énumérée **MODE_REVEIL** contenant les états du réveil (uniquement ceux de la classe **Reveil**.)

Attention la première constante doit posséder la valeur 3

11. Pour terminer, codez la méthode **Controler** de la classe **Reveil** en vous inspirant du diagramme états-transitions réalisé lors de la phase d'analyse.

Pour rappel, cette méthode commence par appeler la méthode **Controler** de la classe **Horloge**, les deux diagrammes à état doivent s'enchaîner.

```
TOUCHES_CLAVIER Revail::Controler(TOUCHES_CLAVIER numTouche)
{
    numTouche = Horloge::Controler(numTouche);

    // Code pour Contrôler le Réveil

    return numTouche;
}
```

Dans le cas parmi représentant le codage du diagramme états-transitions, ajoutez le cas **AUCUN_REGLAGE**, il permettra de lancer la méthodes **SurveillerAlarme** (voir le diagramme de séquence correspondant)

12. Pour la sonnerie on prévoit simplement d'afficher dans un autre cadran le message **ALARME**. Réalisez le codage nécessaire pour cette fonctionnalité. Prévoyez une nouvelle méthode **ArreterAlarme** dont le rôle est d'effacer le message. Complétez le codage du cas **AUCUN_REGLAGE** du réveil pour que l'appui sur la touche **MODE**, arrête l'alarme. (sans changer le mode du réveil)

4. MISE À JOUR DE LA DOCUMENTATION ET DU REPOSITORY GITHUB

13. Mettre à jour votre dossier d'analyse sous Modelio en respectant tous les éléments que vous avez apportés dans le code C++ et en s'assurant de la cohérence entre les diagrammes et le code. Exportez votre projet et placez l'export dans un sous dossier **Analyse** du dossier **ProjetHorloge**.
14. Complétez votre documentation doxygen.
15. Après avoir ajouté le suivi des nouveaux dossiers par Git, remontez tous vos projets du dossier **ProjetHorloge** sur votre GitHub.