

Analyse UML et Codage C++

Projet Reveil sous Linux

~ Séquence n°4 ~

Version 3.0 – janvier 2019

Objectif

- Codage C++
 - Héritage
 - Accès protégé
 - Polymorphisme
 - Résolution de portée
- UML
 - Relations entre classes : Héritage

Conditions de réalisation

Ressources utilisées :

Matériel

Un PC sous linux

Logiciel

NetBeans
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>short</i>	Heure de l'alarme, valeur comprise entre 00 et 11 ou entre 00 et 23 suivant la résolution.
minutesAlarme <i>short</i>	Minute de l'alarme, valeur comprise entre 00 et 59
Alarme <i>booléen</i>	Indique si l'alarme doit être surveillée

Opérations : [Public]

Reveil (<i>in short _nbMode, in short _resolution</i>) par défaut <i>_resolution = 24</i>	
	Constructeur de la classe, initialise les attributs heuresAlame 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

2. MISE A JOUR DE L'ANALYSE UML

1. A partir de l'analyse UML de l'horloge fournie en ressource, importez le projet **Horloge_Prof.zip** et renommez le package **Horloge_Prof** en **Réveil**.
2. Modifiez le diagramme des cas d'utilisation en ajoutant le cas « **Surveiller Alarme** ».
3. Complétez le diagramme de classe en ajoutant la classe **Reveil** et la classe **Sonnerie**. Ajoutez les relations entre les différentes classes
4. .Réalisez le diagramme états-transitions de la classe **Reveil**.
5. Réaliser le diagramme de séquence du cas d'utilisation « Surveiller Alarme »