

Module X9IB010 Programmation Objet pour les biologistes

Description

Mini-projet

Développement d'un logiciel pour un centre de radiologie

Description

L'objectif de ce mini-projet est de développer une application pour un centre de radiologie.

Cette application devra permettre de gérer efficacement les dossiers des patients et d'aider les praticiens au suivi et diagnostic.

Toutes les radiographies possèdent un numéro d'examen unique, un type (radiographie conventionnelle à rayons X, échographie à ultrasons, IRM), le patient concerné, une date de prise, un état (« planifiée » ou « effectuée »), l'ensemble des clichés associée le cas échéant, ainsi qu'un compte-rendu médical textuel protégé par mot de passe. A chaque cliché est associé, un numéro de prise.

Le centre de radiologie souhaite pouvoir gérer l'ensemble des examens au travers d'une seule représentation : une liste de radiographies. Le logiciel doit notamment permettre l'ajout, la suppression, l'affichage des caractéristiques des examens par patient, par date, par numéro d'examen. Le logiciel devra également permettre de charger/sauvegarder la listes des radiographies depuis/vers un fichier.

Travail demandé

Le projet est à effectuer en binômes.

1. Réaliser une analyse de la modélisation objet envisageable pour l'application, en spécifiant notamment les diagrammes de classes.
2. Développer une implémentation de ce logiciel en C++.

Le rapport complet au format pdf ainsi que les fichiers sources commentés sont à envoyer à mourad.oussalah@univ-nantes.fr sous la forme d'UNE SEULE archive dont le nom sera composé des noms de chacun des étudiants du binôme (ex: durantdupont_ProgObjet.zip). On précisera en commentaire dans chaque fichier source les noms et prénoms de chacun des étudiants.

Un projet rendu après la date limite sera affecté d'un malus proportionnel au nombre de jours de retard.

Module X9IB010 Programmation Objet pour les biologistes

Note sur l'évaluation

Le rapport écrit devra comporter :

- le cahier des charges du logiciel (fonctionnalités, etc.),
- une analyse du problème et une présentation de la solution : hiérarchie des classes, définition du comportement des classes.
- une notice utilisateur du logiciel,
- une présentation des résultats du programme (jeux d'essais),
- une conclusion (problèmes rencontrés, améliorations possible)