

Projet individuel de programmation

1 Présentation, principes, déroulement

Projet individuel ; il y a en tout une vingtaine d'heures dans l'emploi du temps, soit 30 à 40h de travail par élève. L'accent est mis davantage sur les aspects « conception logicielle » (au sens documentation, réutilisabilité...) et suivi de projet que sur la difficulté technique du sujet.

Le sujet doit permettre de décomposer le code en deux parties :

1. un ensemble de services (primitives, fonctionnalités de base) : P1
2. un « client » utilisant ces services pour répondre aux besoins exprimés dans le sujet, en utilisant une interface graphique : P2

Dans un premier temps, chaque élève doit implémenter l'ensemble des services demandés (P1). À l'issue de cette première phase, les codes sont échangés entre les élèves ; chacun doit alors récupérer le code qui lui a été attribué (c'est celui d'un autre élève) et le compléter (P2) pour finir le projet. Il y a donc échange de développeur (ou de code selon le point de vue) au milieu du projet.

L'ensemble des services proposés dans la 1ère partie doit bien entendu être défini, mais seules les fonctionnalités doivent être précisées ; l'élève doit pouvoir effectuer des choix d'implémentation (nom et nombre des méthodes, des paramètres, ...). Intérêts :

1. oblige l'élève à documenter son code pour qu'il soit réutilisable par un autre élève
2. chaque élève ayant (a priori) des choix, donc des codes différents, force les élèves en phase 2 à réfléchir aux choix faits par leur fournisseur de 1ère partie

La partie 2 doit utiliser les fonctionnalités fournies en P1 pour satisfaire les exigences du sujet, en mettant en œuvre une interface graphique.

Travail demandé à l'élève

- chaque semaine, l'élève doit fournir un fichier log décrivant en quelques lignes les modifications apportées dans la semaine et une grille d'avancement par fonctionnalité demandée (pas commencé, en cours, fait mais pas testé, fait mais buggé, fini et validé).
- pour chaque partie, l'élève doit fournir une version intermédiaire de son code :
 - pour la partie 1, une version qui tourne, même si toutes les fonctionnalités ne sont pas implémentées (par exemple une fonction booléenne renvoyant systématiquement true) ;
 - pour la partie 2, une version permettant de tester les fonctionnalités de la partie 1, mais sans l'interface.
- à la fin du projet :
 - l'ensemble du code documenté (avec d'éventuelles corrections de la partie 1 reçue par l'élève)
 - une grille d'interévaluation (similaire à ipipip) portant sur la partie 1 reçue par l'élève

2 Planning schématique

