Bachelor 1  
2013/2014

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Projet Algo. et Concepts Objet/C# | | | | |
| CODE | Semestre | Nombre d’heures | Nombre de crédits | Langue |
| EPTR112 | 2 | 40 | 2 | Français |

|  |
| --- |
| Objectifs et compétences |
| Objectifs Assimiler les notions essentielles de la programmation objet : les notions d’encapsulation, de classes, d’héritage et de polymorphisme seront au centre de l’apprentissage. Des notions de maintenabilité et de réutilisabilité de classes seront abordées. Compétences Etre capable de réaliser un programme simple en C#, en mode console et avec WindowsForm, avec découpage modulaire. Etre capable de vérifier la bonne exécution de son programme.  Etre capable d'utiliser les concepts objet dans des programmes utilisant le langage C# - Maintenir des programmes C# - Modifier, Faire évoluer des programmes C# Résumé du projet Vous devez modéliser le fonctionnement (simplifié) d'une banque. Cela vous permet de faire interagir plusieurs types d'objet différents: les comptes, le client, la banque. |
| Formule pédagogique |
| Prérequis EIAD101 – Algorithmique  EIAD102 - Algo et langage C#  EIAD103 - Concepts Objet / Langage C# Déroulement 40 heures Constitution des groupes X groupes de 2 à 4 personnes Livrables Archive au format ZIP comprenant:   * la solution Visual Studio complète ; * un document texte expliquant les étapes de réalisations, les difficultés rencontrées ainsi que la manière dont elles ont été surmontées. |

|  |
| --- |
| Plan détaillé |
| Etape 1 (Compréhension du fonctionnement de la banque) **Le compte**   * Un compte bancaire est identifié par son numéro et par le client à qui il appartient * Il possède un libellé, une date d’ouverture, un solde et un type de compte (Courant, Livret A, LEP, PEL, …) * Un compte courant n’est pas rémunéré alors que Livret l’est. * Un client peut faire un retrait (si son solde le lui permet) ou un dépôt. * Les retraits ne sont pas autorisés sur le PEL avant 5 ans d’ancienneté. * Tous les ans le 31 décembre, les intérêts sont ajoutés aux comptes rémunérés.   **Le client**  Le client est identifié par son numéro est possède les caractéristiques suivantes :   * Nom * Prénom * Adresse * Code postal * Ville * Téléphone * …   **La banque**  La banque permet de gérer les comptes et les clients.  Elle permet, entre autre, de :   * Gérer les clients   + Ajouter un client   + Modifier un client   + Supprimer un client (et tous ses comptes ?)   + Afficher tous les clients   + Trier les clients selon un ordre prédéfini (optionnel) * Gérer les comptes   + Ajouter un compte (si le client existe)   + Modifier un compte   + Supprimer un compte   + Afficher tous les comptes   + Afficher les comptes d’un client   + Trier les comptes selon un ordre prédéfini (optionnel)  Etape 2 (la conception) Vous devez faire un diagramme de classe avant de commencer le développement. Etape 3 (le développement) Implémenter dans l’ordre :   * La classe client * Les classes pour les comptes * La classe pour gérer les clients * La classe pour gérer les comptes * La classe pour gérer la banque   Vous pouvez faire l’application soit en console soit en graphique  Des tests doivent être réalisés tout au long du développement. Etape 4 (Le découpage modulaire) Toutes les classes liées à la banque doivent être dans une librairie spécifique. Soutenance – 1 groupe Remise des livrables  5 minutes de démo du fonctionnement  5 minutes de retour d’expérience  5 minutes de questions/réponses individuelles |
| Annexes | |
|  | |