TD6 – Délégation – A Finaliser 26/04/2019

Objectif

Comprendre la notion de délégation et intégrer cette nouvelle solution dans des exercices globaux

Exercice 1 – Utiliser les délégués déjà créés par C#

Créer une classe Animal_Domestique caractérisée par un nom (string), un pseudo (string) , une date_naissance (Datetime), un poids (double) exprimé en Kg

Créer des instances de cette classe et composer une liste générique (utiliser List<Animal_Domestique>)

Utiliser les méthodes Find, FindAll, Sort, ForEach afin de comprendre l'usage de ces fonctions.

1. Rechercher les chiens de plus de 10kg et afficher les
2. Rechercher un chien avec un pseudo particulier
3. Pour chaque chien, transformer le poids en Pound (/0.4536)
4. Trier les animaux selon le poids, la date de naissance et le nom

Utiliser les 2 écritures (delegate et lambda fonction)

Exercice 2 – Découvrir la création d'un délégué

Une école veut uniformiser le calcul de la note finale pour chaque module, tout en laissant à chaque responsable de module le choix de calculer la moyenne finale de son module comme il le souhaite. Chaque module dispose à la fin d'un tableau de notes de contrôles continus, et d'une note d'examen. La solution retenue est de créer une classe Evaluation, qui contient un tableau des notes des contrôles continus et une note d'examen. Cette classe contient aussi une méthode CalculNoteFinale, qui permet d'obtenir la note finale du module, à partir de la méthode de pondération proposée par l'enseignant dans le main.

1. Coder la classe Evaluation.

2. Compléter le main, en utilisant la classe Evaluation, et en implémentant 3 méthodes pour calculer la note finale :

   a. Moyenne de toutes les notes, y compris celle de l'examen.

   b. Moyenne avec pondération de 40% pour les notes de CC et de 60% pour l'examen

   c. Moyenne avec pondération de 40% pour les notes de CC en éliminant les zéros et de 60% pour l'examen

Exercice 3 – Exercice récapitulatif

Un centre de formation souhaite automatiser la gestion de ses formations à travers la création d'un logiciel. Le logiciel doit faire la distinction entre 3 types de formation. Une formation est identifiée par un nom, et caractérisée par un nombre de participants.

1. Le premier type de formation (formation de longue durée) possède en plus une liste de formateurs.
2. Le deuxième type de formation (formation courte durée), il faut préciser la liste des matières étudiées.
3. Pour le troisième type de formation (formation avec stage) nécessite de préciser la liste des entreprises pouvant accueillir les participants.

Chaque participant possède un numéro d'identification associé à son nom, adresse et téléphone.

Un formateur est identifié par son numéro de sécurité sociale associé à son nom, adresse et téléphone.

Une entreprise est identifiée par son nom, l'adresse de son siège social

Pour une formation de longue durée, le coût est égal à 200 euros par jour de formation.

Pour une formation de courte durée, le coût est égal à 175 euro par matière étudiée.

Pour le troisième type d'intervention, il n'y a pas de coût à facturer aux participants (La formation est financée par les entreprises).

Le centre de formation souhaite

1. Afficher pour chaque formation enregistrée la liste des participants trié par ordre alphabétique.
2. Pour les formations avec stage, afficher aussi la liste des entreprises triée par ordre alphabétique.
3. Une méthode qui calcule le coût d'une formation
4. Rechercher toutes les formations qui ont plus de 10 participants

1. Tracer le diagramme de classe modélisant le programme.
2. Implémenter les classes en C#.
3. Tester le programme.