

Projet POO

BUT2 SD R3.VCOD.10 - Programmation objet

Rendu: 19 octobre

Démarche du projet

Le but de ce projet est de construire un programme reposant sur la notion d'objet ainsi que ses caractéristiques : **Classe**, **Attribut**, **Méthode**, **Héritage**, **Association**, **Agrégation** et **Composition**. Ce projet devra être rédigé en Python. Une attention particulière sera portée à l'utilisation de la programmation orientée objet dans ce projet, au travers de classes et d'objets dédiés facilitant le travail. La partie algorithmique de ce projet est assez légère, et sera guidée et expliquée, il faut porter son attention sur la structure du code, sa portabilité et sa potentielle réutilisabilité ou son extensibilité. C'est une occasion d'implémenter quelques algorithmes tout en mettant en pratique les notions de classes et d'objets vus en cours.

Travail à réaliser

Ce projet va être évalué au travers de trois rendus :

1. le code source (dans un `.zip`) contenant :
 - le(s) fichier(s) de votre programme permettant d'exécuter celui-ci.
 - un fichier `README.md` expliquant comment utiliser votre programme et quelles sont ses éventuelles dépendances (bibliothèque, version, etc.).
2. un rapport (`.pdf`) dans lequel vous :
 - présenterez le déroulement de votre programme.
 - expliquerez l'architecture de votre code et les choix y ayant mené.
 - détaillerez les points critiques/intéressants de votre implémentation.
3. une présentation (`.pdf`) de 10mn, pendant laquelle vous :
 - présenterez le(s) problème(s) posé(s) par votre sujet.
 - expliquerez votre programme et vos choix architecturaux.
 - montrerez, si possible via une démonstration, son fonctionnement.

Le code source, ainsi que le rapport et la présentation, doivent être déposés sur *Chamillo* (sous Travaux) et/ou envoyés à `martin.lentschat@univ-grenoble-alpes.fr` au plus tard le 19 octobre, veille de la soutenance.

Tout apports personnels au projet, comme l'implémentation d'idées enrichissant le projet ou explorant le problème posé, seront les bienvenus. Soyez créatifs!

Sujet I : Embarquement d'un avion

L'ordre d'embarquement des passagers est primordial afin de réduire au minimum le temps nécessaire avant que l'avion ne puisse décoller. Un avion est composé d'un nombre défini de sièges, accessibles via une ou plusieurs allée(s) permettant la progression des passagers. Il existe au minimum deux types de sièges différents : 1^e classe et 2nd classe. Les passagers ont un numéro de siège attribué, chaque siège est reconnaissable par son numéro de colonne et de rangé. Certains passagers emportent avec eux des bagages qu'ils vont souhaiter ranger avant de s'asseoir.

Dans ce projet, vous implémenterez plusieurs méthodes permettant de simuler la progression des passagers, au sein de l'avion, afin que chacun rejoigne sa place. Les sièges et allée(s) seront représentés par des cases. Cette simulation se divisera en tours, et devra respecter les règles suivantes :

- il ne peut y avoir que un passager par case.
- au début de chaque tour, si l'entrée d'une allée est libre, alors un nouveau passager entre dans l'avion.
- chaque passager ne peut se déplacer que d'une case par tour.
- les passagers avec bagages ont besoin d'un tour afin de les ranger.
- le seul cas où un passager peut rentrer dans la case d'un autre est si celui-ci est assis. Cela demande alors au passager debout un tour supplémentaire.

Afin de rechercher la méthode permettant de faire embarquer les passagers le plus rapidement, vous testerez *à minima* deux des algorithmes suivant :

- l'ordre des passagers est choisi aléatoirement.
- les passagers de 1^e classe sont appelés à embarquer en premier.
- les numéros de rangés des sièges sont utilisées afin d'appeler les passagers dans un certain ordre.
- les numéros de colonnes des sièges sont utilisées afin d'appeler les passagers dans un certain ordre.

Le but est donc ici d'implémenter cette simulation en exploitant les objets et leurs fonctionnalités. Votre présentation et votre rapport devront décrire les choix faits et contenir une petite analyse des résultats de votre travail. Les consignes ci-dessus devront être respectées, mais tout ajouts, ou modifications intéressantes / enrichissant votre compréhension du sujet seront valorisés.

Projet POO

BUT2 SD R3.VCOD.10 - Programmation objet

Rendu: 19 octobre

Démarche du projet

Le but de ce projet est de construire un programme reposant sur la notion d'objet ainsi que ses caractéristiques : **Classe**, **Attribut**, **Méthode**, **Héritage**, **Association**, **Agrégation** et **Composition**. Ce projet devra être rédigé en Python. Une attention particulière sera portée à l'utilisation de la programmation orientée objet dans ce projet, au travers de classes et d'objets dédiés facilitant le travail. La partie algorithmique de ce projet est assez légère, et sera guidée et expliquée, il faut porter son attention sur la structure du code, sa portabilité et sa potentielle réutilisabilité ou son extensibilité. C'est une occasion d'implémenter quelques algorithmes tout en mettant en pratique les notions de classes et d'objets vus en cours.

Travail à réaliser

Ce projet va être évalué au travers de trois rendus :

1. le code source (dans un `.zip`) contenant :
 - le(s) fichier(s) de votre programme permettant d'exécuter celui-ci.
 - un fichier `README.md` expliquant comment utiliser votre programme et quelles sont ses éventuelles dépendances (bibliothèque, version, etc.).
2. un rapport (`.pdf`) dans lequel vous :
 - présenterez le déroulement de votre programme.
 - expliquerez l'architecture de votre code et les choix y ayant mené.
 - détaillerez les points critiques/intéressants de votre implémentation.
3. une présentation (`.pdf`) de 10mn, pendant laquelle vous :
 - présenterez le(s) problème(s) posé(s) par votre sujet.
 - expliquerez votre programme et vos choix architecturaux.
 - montrerez, si possible via une démonstration, son fonctionnement.

Le code source, ainsi que le rapport et la présentation, doivent être déposés sur *Chamillo* (sous Travaux) et/ou envoyés à `martin.lentschat@univ-grenoble-alpes.fr` au plus tard le 19 octobre, veille de la soutenance.

Tout apports personnels au projet, comme l'implémentation d'idées enrichissant le projet ou explorant le problème posé, seront les bienvenus. Soyez créatifs!

Sujet II : Créateur de prénoms

Trouver un prénom à donner à un nouveau-né n'est pas toujours facile. Dans ce projet, nous voudrions pouvoir implémenter une méthode permettant de créer de nouveaux prénoms en se basant sur les règles de constructions de prénoms existants. Un corpus de prénoms provenant de différentes langues, réelles et fictives, sera fourni. Dans ces fichiers, chaque ligne comprend un symbole indiquant le genre du prénom (m : masculin, f : féminin, x : neutre/épïcène) et le prénom, séparés par une espace.

Les prénoms, comme tout autre mot, sont composés de lettres. Ces lettres forment des syllabes, définies comme des ensembles de lettres répondants aux règles suivantes :

- le noyau d'une syllabe est toujours une voyelle unique.
- si possible, la syllabe inclut une consonne d'accroche devant son noyau.
- les consonnes après le noyau, à l'exception de l'éventuelle consonne d'accroche de la syllabe suivante, forment la queue de la syllabe.
- au sein d'un même mot, une même lettre ne peut appartenir qu'à une syllabe.

Afin de créer de nouveaux prénoms, vous vous baserez sur cette structure en syllabe. L'objectif est d'identifier et utiliser les règles tacites de construction des prénoms dans chaque langue. Votre programme devra ainsi calculer, pour chaque syllabe, la probabilité que celle-ci apparaisse au début d'un prénom, la probabilité d'apparition avant et/ou après d'autres syllabes, ainsi que la probabilité d'être la syllabe finale. Cela vous permettra alors de générer de nouveaux prénoms en :

- sélectionnant une ou plusieurs langues, ainsi que un ou plusieurs genres.
- tirant au hasard une syllabe de début possible selon ces paramètres.
- ajoutant de nouvelles syllabes suivant les règles de probabilité extraites précédemment.

Le but est donc ici d'implémenter cette simulation en exploitant les objets et leurs fonctionnalités. Votre présentation et votre rapport devront décrire les choix faits et contenir une petite analyse des résultats de votre travail. Les consignes ci-dessus devront être respectées, mais tout ajouts, ou modifications intéressantes / enrichissant votre compréhension du sujet seront valorisés.

Projet POO

BUT2 SD R3.VCOD.10 - Programmation objet

Rendu: 19 octobre

Démarche du projet

Le but de ce projet est de construire un programme reposant sur la notion d'objet ainsi que ses caractéristiques : **Classe**, **Attribut**, **Méthode**, **Héritage**, **Association**, **Agrégation** et **Composition**. Ce projet devra être rédigé en Python. Une attention particulière sera portée à l'utilisation de la programmation orientée objet dans ce projet, au travers de classes et d'objets dédiés facilitant le travail. La partie algorithmique de ce projet est assez légère, et sera guidée et expliquée, il faut porter son attention sur la structure du code, sa portabilité et sa potentielle réutilisabilité ou son extensibilité. C'est une occasion d'implémenter quelques algorithmes tout en mettant en pratique les notions de classes et d'objets vus en cours.

Travail à réaliser

Ce projet va être évalué au travers de trois rendus :

1. le code source (dans un `.zip`) contenant :
 - le(s) fichier(s) de votre programme permettant d'exécuter celui-ci.
 - un fichier `README.md` expliquant comment utiliser votre programme et quelles sont ses éventuelles dépendances (bibliothèque, version, etc.).
2. un rapport (`.pdf`) dans lequel vous :
 - présenterez le déroulement de votre programme.
 - expliquerez l'architecture de votre code et les choix y ayant mené.
 - détaillerez les points critiques/intéressants de votre implémentation.
3. une présentation (`.pdf`) de 10mn, pendant laquelle vous :
 - présenterez le(s) problème(s) posé(s) par votre sujet.
 - expliquerez votre programme et vos choix architecturaux.
 - montrerez, si possible via une démonstration, son fonctionnement.

Le code source, ainsi que le rapport et la présentation, doivent être déposés sur *Chamillo* (sous Travaux) et/ou envoyés à `martin.lentschat@univ-grenoble-alpes.fr` au plus tard le 19 octobre, veille de la soutenance.

Tout apports personnels au projet, comme l'implémentation d'idées enrichissant le projet ou explorant le problème posé, seront les bienvenus. Soyez créatifs!

Sujet III : Fourmilière

Une fourmilière est un système complexe s'intégrant dans un environnement vaste. Les fourmis cherchent en effet à se reproduire et à étendre la fourmilière, mais doivent faire attention à ne pas consommer toutes les ressources à leur disposition ou risquer de disparaître.

Votre simulation se déroulera sur plusieurs années, divisées en saisons. Votre fourmilière débutera avec un nombre choisi de fourmis. Il existe des fourmis garde défendant la fourmilière face aux menaces, des récolteuses ramenant les ressources de l'environnement dans la fourmilière et des puéricultrices élevant de nouvelles fourmis. Lors de chaque saisons, la quantité de ressources disponibles dans l'environnement change en fonction de la quantité actuelle :

- le printemps voit les ressources de l'environnement doubler.
- l'été, la quantité de ressources augmente de 50%.
- à l'automne, les ressources ne changent pas.
- l'hiver, les ressources diminuent de 25%.

De nombreux paramètres de départ (nombre de fourmis et ressources de départ) ainsi que les critères des algorithmes déterminant le comportement des fourmis (quand récolter, quand élever de nouvelles fourmis, comment attribuer les rôles, etc.) sont laissés à votre discrétion, vous êtes encouragés à explorer et à inventer les vôtres. Prenez le temps d'observer l'évolution de votre fourmilière et de jouer avec vos paramètres. Cependant, certaines règles s'appliquent sur les fourmis :

- les fourmis ont une durée de vie d'un an.
- à chaque saison, un nombre aléatoire d'attaques atteint la fourmilière. Chaque attaque non contrecarrée par une fourmi garde tue alors une fourmi (garde ou autre).
- à chaque saison, les fourmis mangent une partie des ressources stockées. Les gardes consomment plus que les récolteuses, qui consomment plus que les puéricultrices.
- une unique fourmis puéricultrice prend un cycle à élever une unique nouvelle fourmi, cette fourmi juvénile doit également manger.

Le but est donc ici d'implémenter cette simulation en exploitant les objets et leurs fonctionnalités. Votre présentation et votre rapport devront décrire les choix faits et contenir une petite analyse des résultats de votre travail. Les consignes ci-dessus devront être respectées, mais tout ajouts, ou modifications intéressantes / enrichissant votre compréhension du sujet seront valorisés.

Projet POO

BUT2 SD R3.VCOD.10 - Programmation objet

Rendu: 19 octobre

Démarche du projet

Le but de ce projet est de construire un programme reposant sur la notion d'objet ainsi que ses caractéristiques : **Classe**, **Attribut**, **Méthode**, **Héritage**, **Association**, **Agrégation** et **Composition**. Ce projet devra être rédigé en Python. Une attention particulière sera portée à l'utilisation de la programmation orientée objet dans ce projet, au travers de classes et d'objets dédiés facilitant le travail. La partie algorithmique de ce projet est assez légère, et sera guidée et expliquée, il faut porter son attention sur la structure du code, sa portabilité et sa potentielle réutilisabilité ou son extensibilité. C'est une occasion d'implémenter quelques algorithmes tout en mettant en pratique les notions de classes et d'objets vus en cours.

Travail à réaliser

Ce projet va être évalué au travers de trois rendus :

1. le code source (dans un `.zip`) contenant :
 - le(s) fichier(s) de votre programme permettant d'exécuter celui-ci.
 - un fichier `README.md` expliquant comment utiliser votre programme et quelles sont ses éventuelles dépendances (bibliothèque, version, etc.).
2. un rapport (`.pdf`) dans lequel vous :
 - présenterez le déroulement de votre programme.
 - expliquerez l'architecture de votre code et les choix y ayant mené.
 - détaillerez les points critiques/intéressants de votre implémentation.
3. une présentation (`.pdf`) de 10mn, pendant laquelle vous :
 - présenterez le(s) problème(s) posé(s) par votre sujet.
 - expliquerez votre programme et vos choix architecturaux.
 - montrerez, si possible via une démonstration, son fonctionnement.

Le code source, ainsi que le rapport et la présentation, doivent être déposés sur *Chamillo* (sous Travaux) et/ou envoyés à `martin.lentschat@univ-grenoble-alpes.fr` au plus tard le 19 octobre, veille de la soutenance.

Tout apports personnels au projet, comme l'implémentation d'idées enrichissant le projet ou explorant le problème posé, seront les bienvenus. Soyez créatifs!

Sujet IV : Le Corbeau du village

Dans un petit village, une personne mal intentionnée (le Corbeau) distribue des lettres anonymes dénonçant les secrets des uns et des autres. L'objectif de ce projet est de simuler le village, ainsi que les évolutions des relations entre personnes dans cette situation.

Les maisons du village sont chacune occupée par une famille, composée d'au moins deux personnes qui ne savent pas que un corbeau habite dans leur village (celles-ci sont considérées comme ignorantes). Chaque maison est voisine d'au minimum une autre maison. Régulièrement, le corbeau distribue une nouvelle lettre adressée à une personne spécifique, n'importe où dans le village. Les membres de la famille peuvent alors avoir plusieurs réactions :

- la personne ciblée prend toujours peur, et ne recherchera alors pas le Corbeau.
- les autres membres de la famille ont trois réactions possibles :
 - la plupart restent ignorants, ne sachant pas ce qui arrive à un membre de leur famille, ils n'adoptent pas de comportement spécifique.
 - parfois certains accusent immédiatement une personne d'une maison proche. Si celle-ci n'est pas le Corbeau, alors ces deux personnes sont brouillées et refuseront de collaborer dans l'avenir.
 - d'autres deviennent suspicieux, et vont alors régulièrement enquêter sur l'un de leur voisin pour rechercher le Corbeau. Plusieurs possibilités existent alors :
 - si la personne suspectée est le Corbeau, celui-ci est arrêté.
 - si la personne est ignorante, alors elle le reste.
 - si l'autre personne est également suspicieuse, alors un groupe d'enquêteurs se forme (ou est rejoint si ce groupe existe déjà). Les personnes de ce groupe se soutiennent mutuellement et ne peuvent alors plus être intimidées par le Corbeau.

Les personnes suspicieuses se rendent en priorité aux maisons les plus proches, la distance étant définie par les relations de voisinage entre maisons.

Le but est donc ici d'implémenter cette simulation en exploitant les objets et leurs fonctionnalités. Votre présentation et votre rapport devront décrire les choix faits et contenir une petite analyse des résultats de votre travail. Les consignes ci-dessus devront être respectées, mais tout ajouts, ou modifications intéressantes / enrichissant votre compréhension du sujet seront valorisés.