

TP jeu des maladies

Le jeu des maladies consiste à gérer l'état de patients malades. Dans cette simulation à chaque *tour*, un an se passera et la maladie sera plus ou moins virulente tandis que l'état du patient empirera ou s'améliorera.

Votre mission si vous l'acceptez est de combattre les maladies afin que vos patients continuent de vivre des jours heureux !

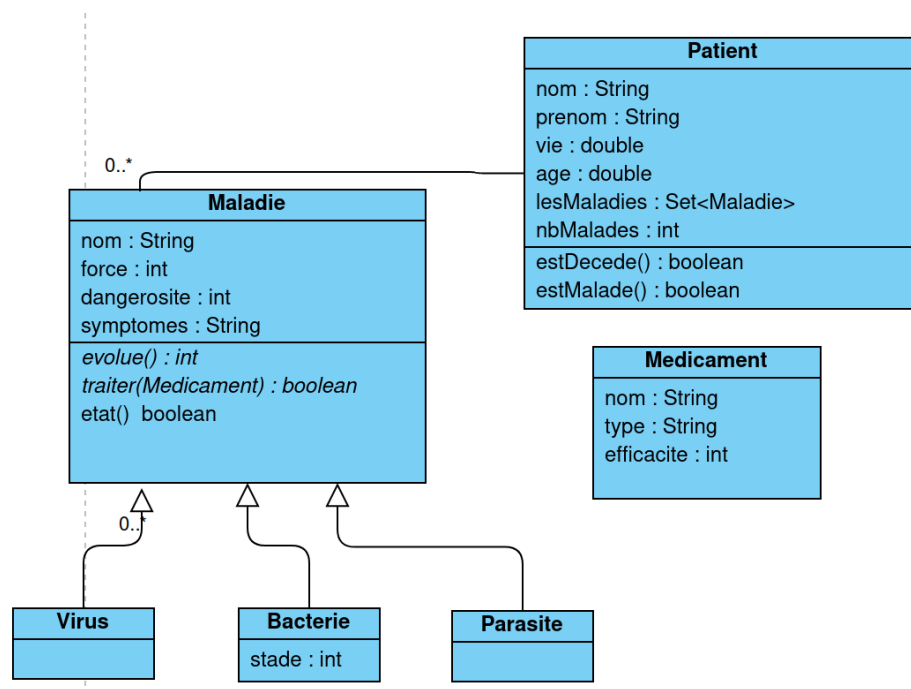
Au début du jeu, différents patients seront créés et attraperont une/des maladie(s) au hasard (3 au maximum). Le joueur, en tant que médecin fera tour pour les soigner.

Voici le déroulement d'1 tour de jeu pour chaque patient :

1. Affichage d'un message en fonction de l'état du patient (décédé, sain, malade).
2. Le médecin choisit le médicament qui composera le traitement du patient pour toute l'année.
3. Le médicament est appliqué à toutes les maladies, celles qui ont été éradiquées sont supprimées.
4. Les maladies qui sont toujours actives évoluent et font perdre des points de vie au patient
5. Le patient vieillit d'un an.

Le jeu se termine lorsque tous les patients sont sains ou si l'un d'entre eux décède.

Partie 1 – Création des classes du jeu



1. Création de la classe Medicament

La classe Medicament représente les différents médicaments qui pourront être utilisés pour combattre les maladies. La classe Medicament contient les informations suivantes :

Attributs :

- nom (String)
- type (String) doit obligatoirement faire partie de la liste suivante : Antibiotique, Antiviral, Antiparasite. On pourra alors soit utiliser un String, soit une enum.

Constructeur :

- un constructeur par paramètres

Méthodes :

- Les getters et setters que vous pensez nécessaires.
- La méthode toString qui retourne une représentation du médicament sous forme de chaîne de caractères.

2. Création des classes Maladie, Virus, Bacterie, Parasite

Créez la classe abstraite *Maladie* à partir des informations suivantes :

Attributs :

- Nom : (String) le nom de la maladie
- force : (int) la force de la maladie, lorsque la force atteint 0, la maladie est éradiquée.
- Dangersité : (int) la sévérité de la maladie (influe sur la vie du patient) [0,10]
- Symptômes : (String) les effets de la maladie sur le corps

Constructeurs :

- Un constructeur avec tous les arguments pour définir chacun des attributs.

Méthodes :

- Etat : boolean retourne true si la maladie est éradiquée, false sinon.
- La méthode toString qui retourne une représentation de la maladie sous forme de chaîne de caractères.
- Méthode abstraite evolue() : int : la maladie se développe et retourne le nombre de points de vie à retirer au patient.
- Méthode abstraite : traiter(Medicament med) : boolean applique le traitement à la maladie et retourne true si la maladie est éradiquée, false sinon.

Créer les classes filles Virale, Bacterie et Parasite qui héritent de la classe Maladie.

Chacune de ces classes devra implémenter les méthodes abstraites *evolue* et *traiter* selon leurs spécificités.

Pour une maladie Virus, l'utilisation d'un médicament de type antiviral fait perdre 10 points à la maladie, l'utilisation d'un médicament de type antibio fait perdre 1 point, les autres n'ont aucun effet.

La maladie évolue en augmentant sa force de 2 points par tranche de 25 points de force. L'impact sur le patient correspond à un dixième de la force de la maladie, multiplié par la dangerosité.

Maladie Bacterie

Une maladie de type Bacterie possède un stade de maladie : (short) : [1,4] : bénin, modéré, dangereux, sévère (on pourra également utiliser une *enum* dans un second temps). Ce stade sera déterminé aléatoirement de manière uniforme lors de la création de l'objet (donc dans le constructeur de Bacterie).

Selon le stade de la maladie, le médicament a un impact différent :

- Stade bénin : un antibiotique éradique la maladie, un antiviral fait perdre 10 points.
- Stade modéré : un antibiotique fait perdre 10 points à la maladie, un antiviral fait perdre 3 points.
- Stade dangereux : un antibiotique fait descendre la maladie au stade modéré.
- Stade sévère : un antibiotique fait perdre 5 points à la maladie.

La maladie évolue aléatoirement : soit elle gagne 5 points, soit elle augmente d'un stade. Le nombre de points de vie perdus par le patient dépend du stade de la maladie et sera arrondi à l'entier le plus proche.

- Au stade bénin, le patient perdra 20% de la force de la maladie
- Au stade modéré, le patient perdra 30% de la force de la maladie
- Au stade dangereux, le patient perdra 70% de la force de la maladie
- Au stade sévère, le patient perdra 90% de la force de la maladie

Maladie Parasite

Pour une maladie de type Parasite, l'utilisation d'un antiparasitaire détruit la maladie totalement, les autres médicaments retirent 20 points.

La maladie n'évolue pas, elle fait perdre 3 points de vie au patient.

3. Création de la classe Patient

La classe Patient représente les patients qu'il faudra soigner dans le jeu.

Attributs :

- Prénom (String)
- Nom (String)
- Vie : (*double*) la 'santé' du patient
- Age (double) l'âge influera sur le système d'immunitaire du patient
- lesMaladies : un HashSet comportant toutes les maladies du patient (le type de collection HashSet permet d'éviter les doublons)

Attribut de classe :

- nbMalades correspond au nombre de patients encore malades (un patient est malade à partir du moment où il est affecté par une maladie)

Constructeur :

- Patient (String prenom, String nom, double vie, double age)

Méthodes :

- Les Getters et Setters que vous pensez nécessaires
- ajoutMaladie(Maladie m) : void : méthode qui affecte une maladie au patient
- estMalade() : boolean : méthode qui retourne false si le patient n'a pas de maladie, true sinon.
- estDecede() : boolean : méthode qui retourne vrai si et seulement si le patient est décédé.
- seSoigner(Medicament med) : void : le médicament med agit sur chaque maladie du patient, si une maladie est éradiquée à l'issue du traitement, elle est supprimée du HashSet. Si toutes les maladies sont éradiquées en fin de traitement, le patient n'est plus malade.
- subirMaladie() : boolean : Les maladies du HashSet évoluent et attaquent le patient, cela peut avoir pour effet de diminuer son nombre de points de vie. L'âge influera sur l'impact de la maladie selon les règles suivantes :
 - impact augmenté d'un quart pour les moins de 15 ans,
 - impact augmenté d'un tiers pour les plus de 65 ans
 - sinon pas d'augmentation sur l'impact

4. Test des classes

Créer une classe de Test contenant un main afin de valider la création des classes Medicament, Patient, Maladie, Virale, Bacterie, Parasite.

Partie 2 : création du jeu

Initialisation

Créer une classe InitJeu qui permettra d'initialiser les objets nécessaires au jeu des maladies. Vous trouverez quelques exemples de Maladie et Medicament en annexe.

- Créer une méthode de classe initMaladie qui retourne une liste de Maladie.
- Créer une méthode de classe initMedicament qui retourne une liste de Medicament.
- Créer une méthode de classe initPatient qui retourne une liste de Patient. Chaque patient sera atteint de 1 à 3 maladies affectées de manière aléatoire.

Moteur de jeu

Créer la classe Jeu qui s'occupera de gérer le moteur du jeu à travers la liste des patients malades. La première étape consiste à initialiser le jeu à l'aide des méthodes de la classe InitJeu.

Voici le déroulement d'un tour de jeu pour chaque patient :

1. Affichage d'un message en fonction de l'état du patient (décédé, sain, malade).
2. Le médecin choisit le médicament qui composera le traitement du patient pour toute l'année.

3. Le médicament est appliqué à toutes les maladies, celles qui ont été éradiquées sont supprimées.
4. Si les maladies sont toujours actives, elles évoluent, ce qui augmente leur force, et fait baisser le nombre de points de vie du patient (en fonction de ce qui est retourné par la méthode evolue)

Le jeu se termine lorsque tous les patients sont sains ou si l'un d'entre eux décède.

Documentation technique

Créer la JavaDoc des classes du jeu.

Compléter les commentaires javadoc pour chacune des classes Maladie, Virus, Bacterie, Parasite, Medicament, Patient :

- Présentation de la classe
- Constructeur
- Méthodes

Générer la JavaDoc du projet.

Parasites :

<i>Nom</i>	<i>Vie</i>	<i>Force</i>	<i>Dangerosité</i>	<i>Effets</i>
Poux	20	1	1	Démangeaisons au niveau des cheveux
Gale	50	4	6	Importantes démangeaisons au niveau de la peau
Ténia	150	6	4	Perte importante de poids, Maux de ventre violents
Malaria	300	9	7	Fièvre élevée soudaine, en poussées ou constante

Exemple de médicaments

<i>Nom</i>	<i>Type</i>
Amoxicilline	Antibiotique
Bristopen	Antibiotique
Clamoxyl	Antibiotique
Orbénine	Antibiotique
Prioderm	Antiparasitaire
Ascabiol	Antiparasitaire
Fluvermal	Antiparasitaire
Combantrin	Antiparasitaire
Aciclovir	Antiviral
Iduviran	Antiviral