

Dans ce TP, nous vous proposons de créer un petit hôpital, puis de le gérer.

L'hôpital contient des médecins, qui servent d'interface entre l'hôpital et vous (l'utilisateur du programme). Au travers de ces médecins, vous pouvez soigner les patients, leur rendre visite, ...

Ci-dessous nous vous donnons les spécifications minimales du programme.

Elles sont parfois incomplètes. A vous de créer des classes, des variables d'instance, des méthodes ou des interfaces supplémentaires ou autre dès que cela semble nécessaire. On utilisera les classes Patient, FemmeEnceinte, Médecin, Obstétricien et Chambre déjà écrites lors d'un TD précédent.

Les médecins

L'hôpital décrit comporte au minimum des services d'Obstétrique, de Chirurgie et de Diagnostic.

Au minimum, votre programme doit pouvoir gérer des médecins qui soient : obstétriciens, internistes, des chirurgiens orthopédiques et chirurgiens thoraciques.

Chaque médecin peut suivre plusieurs patients (penser au tableau de patients), mais les charges n'étant pas les mêmes pour chaque type de patient, le nombre maximal de patients par spécialité est différent: les obstétriciens peuvent suivre jusqu'à 4 patients, les chirurgiens 5 et les internistes 1 seul.

Les médecins sont au moins définis par leur nom et prénom, âge et sexe. Une méthode doit exister pour pouvoir afficher les caractéristiques des médecins, incluant si cela est choisi la liste des patients qu'il suit.

Un médecin doit pouvoir rendre visite à un patient. Il lui est impossible de rendre visite à un patient qui n'est pas de son service.

On utilisera une interface Operer pour regrouper les méthodes et caractéristiques de chaque type d'opération selon la catégorie de médecin selon les quelques règles suivantes.

L'interface comprend une méthode « anesthésier », une méthode « operer », et toute autre méthode que vous jugerez utile.

Lorsqu'un chirurgien effectue une opération cardio-thoracique, il ouvre la cage thoracique, traite le problème dans la cage thoracique, puis referme la cage thoracique, sous anesthésie générale.

Pour une opération orthopédique, il doit placer une prothèse sous anesthésie générale. Pour un accouchement, l'opération est une césarienne sous anesthésie locale.

Préalablement à chaque opération, le patient doit être endormi. Si le patient est allergique, on utilisera un produit anesthésique A, sinon un produit B.

L'hôpital

Il reste maintenant à concevoir l'hôpital. Un hôpital possède un nom, plusieurs médecins, un nombre maximal de chambres nbMax et un tableau de nbMax chambres. Les méthodes d'un hôpital permettent

1. d'afficher la liste des patients de toutes les chambres,
2. d'afficher la liste des patients présents au bloc opératoire, en salle de réveil et en salle de travail.
3. et d'afficher le nombre de patients présents dans l'hôpital, en salle de réveil, en salle de travail et au bloc opératoire.

Applet

Il reste maintenant à réaliser une applet très simple permettant d'interfacer graphiquement une partie de votre programme «hôpital».

L'applet devra afficher la gestion d'un patient par le médecin interne Gregory House. La date d'entrée du patient dans le service du Dr House est choisie par l'utilisateur (attention à gérer tous les cas critiques de dates...).

Afin d'effectuer son diagnostic, une liste de 10 médicaments est proposée au Dr House sur l'interface graphique. Un objet graphique permet d'afficher en permanence l'état du patient (guéri ou malade). Seul un seul des médicaments proposés permet de guérir le patient, il sera choisi aléatoirement par l'ordinateur et n'est pas connu de l'utilisateur. Lorsque l'utilisateur choisit un mauvais médicament, celui-ci disparaît de la liste proposée à l'utilisateur sur l'interface graphique. Lorsque l'utilisateur choisit le bon médicament, le patient est guéri. La date de sortie du service est alors calculée selon la règle : 1 médicament testé = 1 jour d'hospitalisation, la facture est présentée au patient sur l'interface graphique et le Dr House peut traiter le patient suivant.

L'initialisation de l'applet devra en particulier créer les objets nécessaires au bon fonctionnement de cette applet.