

TP n°6

Le menu

Dans la suite de la programmation de notre cas d'étude, nous allons créer depuis la console un menu. Ce menu va nous permettre de gérer :

- Les hôtes,
- Les logements,
- Les voyageurs
- Et les réservations.

Pour chaque élément on va pouvoir réaliser les actions suivantes :

- Lister,
- Ajouter,
- Supprimer.

L'objectif est d'obtenir le résultat suivant :

```
Bienvenue chez AirBnB
-----
Saisir une option :
1 : Liste des hôtes
2 : Liste des logements
3 : Liste des voyageurs
4 : Liste des réservations
5 : Fermer le programme
1
-----
Liste des hôtes
n°0 : Peter Bardu (28 ans) qui s'engage à répondre dans les 12 heures.
n°1 : Patrick Martin (32 ans) qui s'engage à répondre dans les 12 heures.
Saisir une option :
1 : Ajouter un hôte
2 : Supprimer un hôte
3 : Retour
3
-----
Saisir une option :
1 : Liste des hôtes
2 : Liste des logements
3 : Liste des voyageurs
4 : Liste des réservations
5 : Fermer le programme
2
-----
Liste des logements
```

```

n°0 : Peter Bardu (28 ans) qui s'engage à répondre dans les 12 heures.
Le logement est une maison située 81 Rue Colbert, 37000 Tours.
Superficie : 140m2
Superficie Jardin : 500m2
Piscine : oui
Saisir une option :
1 : Ajouter un logement
2 : Supprimer un logement
3 : Retour
1
-----
Ajouter un nouveau logement
Saisir une option :
1 : Ajouter une maison
2 : Ajouter un appartement
3 : Retour
1
-----
Ajouter une maison
Liste des hôtes :
n°0 : Peter Bardu (28 ans) qui s'engage à répondre dans les 12 heures.
n°1 : Patrick Martin (32 ans) qui s'engage à répondre dans les 12 heures.
Numéro de l'hôte : 1
Tarif journalier : 56
Adresse : 146 Rue George Sand, 59553 Cuincy
Superficie : 120
Nombre de voyageurs max : 4
Superficie du jardin : 3000
Piscine (0 : non, 1 : oui) : 0

Votre maison a été ajouté avec succès
-----
Liste des logements
n°0 : Peter Bardu (28 ans) qui s'engage à répondre dans les 12 heures.
Le logement est une maison située 81 Rue Colbert, 37000 Tours.
Superficie : 140m2
Superficie Jardin : 500m2
Piscine : oui
n°1 : Patrick Martin (32 ans) qui s'engage à répondre dans les 12 heures.
Le logement est une maison située 146 Rue George Sand, 59553 Cuincy.
Superficie : 120m2
Superficie Jardin : 3000m2
Piscine : non
Saisir une option :
1 : Ajouter un logement
2 : Supprimer un logement
3 : Retour
3
-----
Saisir une option :
1 : Liste des hôtes
2 : Liste des logements
3 : Liste des voyageurs

```

```
4 : Liste des réservations
5 : Fermer le programme
5
A bientôt
```

1° - Pour commencer, créez un nouveau sous package dans lequel nous allons avoir des classes de gestion du menu.

Nom du package : menu

2° - Créez dans ce package une classe Menu avec une méthode main.

```
public class Menu {

    static Scanner scanner;

    public static void main(String[] args) {

        System.out.println("Bienvenue chez AirBnB");

        scanner = new Scanner(System.in);
        // scanner.useDelimiter("\n");

        ...

        scanner.close();

    }

}
```

3° - Dans cette classe, créez une méthode listerMenu() qui ne renvoie rien. Déclarez cette méthode package et static.

4° - Implémentez cette méthode pour obtenir le résultat suivant :

```
-----
Saisir une option :
1 : Liste des hôtes
2 : Liste des logements
3 : Liste des voyageurs
4 : Liste des réservations
5 : Fermer le programme
```

Utilisez l'objet scanner pour demander le choix à l'utilisateur.

Que se passe-t-il si la valeur est supérieur à 5 ou si l'utilisateur rentre une valeur décimale ou une chaîne de caractères ?

1° - Les exceptions

1.1 - Block try/catch

Créez une méthode choix(int maxValue) qui retourne une valeur entière saisie par l'utilisateur.

Cette valeur doit être obligatoirement comprise entre 1 et maxValue.

Si cette valeur n'est pas un entier ou est un entier non compris entre 1 et maxValue redemandez à l'utilisateur sa saisie autant de fois qu'il le faudra.

Utilisez un block try/catch afin d'intercepter une potentielle exception.

1.2 - Méthode avec throws

Nous allons créer une nouvelle classe qui va nous permettre de gérer nos hôtes (GestionHotes).

1° - Créer une méthode que l'on va appeler listerHotes() de la façon suivante :

```
public class GestionHotes {  
  
    static void listerHotes() {  
  
        System.out.println("-----");  
        System.out.println("Liste des hôtes ");  
  
        System.out.println("Saisir une option : ");  
        System.out.println("1 : Ajouter un hôte");  
        System.out.println("2 : Supprimer un hôte");  
        System.out.println("3 : Retour");  
  
        switch (Menu.choix(3)) {  
            case 1:  
                ajouterHote();  
                break;  
            case 2:  
                supprimerHote();  
                break;  
            case 3:  
                Menu.listerMenu();  
                break;  
        }  
    }  
}
```

2° - Dans cette classe nous allons créer une méthode ajouterHote() qui va nous permettre de créer un nouvel hôte depuis une saisie dans la console.

Utilisez le scanner de Menu pour saisir le nom, le prénom et l'âge (Menu.scanner).

3° - Ajoutez à votre méthode `throws Exception`, cette mention indique que le code de cette méthode peut lever une exception.

```
static void ajouterHote() throws Exception {  
    ...  
}
```

4° - Gérez l'exception, à l'aide d'un block try/catch, lorsque vous appelez cette méthode.

5° - Si tout se passe bien, affichez le nouvelle hôte à la fin de la méthode.

2° - Les listes

Notre modèle ne va pas très loin si nous ne sauvegardons pas nos instances dans des listes. Nous allons créer une liste d'hôtes depuis le Menu afin de pouvoir réaliser des actions dessus comme l'affichage, l'ajout ou la suppression d'un élément.

1° - Dans la classe Menu, déclarez un attribut membre de type `ArrayList<Hote>`. Initialisez cette attribut dans la fonction main avec le constructeur par défaut d'`ArrayList`. Notre liste sera par conséquent vide.

2° - Dans la méthode `ajouterHote()` de la classe `GestionHote`, ajoutez à votre liste l'hôte qui est créé. Utilisez pour cela la méthode `add()` de votre objet `ArrayList`.

3° - De façon similaire créez maintenant une méthode `supprimerHote()` qui supprime l'hôte de la liste depuis son indice dans la liste. Déclarez la méthode avec `throws` car si la valeur rentrée n'est pas entière ou que l'indice n'existe pas dans la liste, une exception sera levée. "Catcher" cette erreur de la même manière que pour l'ajout.

3° - Pour aller plus loin

Le constructeur de la classe `Reservation` est également capable de lever une exception. On part du principe que si le contrat d'un séjour n'est pas respecté, il n'est pas possible de créer une réservation. Dans ce cas, le constructeur de `Reservation` lèvera une exception ce qui ne créera pas d'instance.

1° - Ajoutez dans le constructeur de `Reservation` les tests de validation. Si un test ne passe pas, alors créez une nouvelle exception à l'aide du mot clé `new`.

2° - Ajoutez `throws` à votre constructeur, gérez la capture de l'exception.

4° - Créez un fichier

Enregistrez dans un fichier au format texte une réservation lors de la création.

Ecrire les informations suivantes :

```
Numéro du voyageur : 1  
Numéro du logement : 3  
Date d'arrivée (DD/MM/YYYY) : 12/12/2016  
Nombre de nuits : 4  
Nombre de personnes : 2
```