

Nom : \_\_\_\_\_

Prénom : \_\_\_\_\_

Identifiant : \_\_\_\_\_ Groupe : \_\_\_\_\_ Enseignant : \_\_\_\_\_

/20



Haute École Bruxelles-Brabant  
École Supérieure d'Informatique  
Bachelor en Informatique

mars 2024  
DEV2  
Equipe dev2

## DEV2 – Développement II

### Bilan

#### *Bilan – dev2*

#### Consignes

1. Le bilan dure 1h30 et se fait sur une machine du réseau pédagogique.
2. Vous n'avez pas accès à l'Internet
3. Vous pouvez utiliser vos fichiers disponibles localement et vos notes papier.
4. Avant de commencer, clonez le dépôt (sur Gitesi) qui a été créé pour vous.
5. Tous les fichiers devront être placés dans le clone, créé ci-dessus.
6. Lorsque vous avez fini, créez un commit contenant vos fichiers. Depuis le clone :
  - ▷ `git status`
  - ▷ `git add .`
  - ▷ `git commit -m "remise examen"` (La première commande sert à vérifier que tout est là, et rien d'autre.)
7. Poussez le commit vers le dépôt (sur Gitesi) avec `git push`

Vous allez écrire un programme simulant un réseau informatique domestique (avec plusieurs ordinateurs qui se connectent à un unique routeur-modem).

Créez un projet nommé `bilan2024` dans votre clone. Toutes les classes seront dans un unique package nommé `g12345.bilan` (où vous remplacez 12345 par votre matricule).

- 1 Créez une classe `Computer` (n'oubliez pas le package!), qui représente une machine sur le réseau, et ajoutez lui deux attributs :

- ▷ une chaîne de caractères `name` et
- ▷ un entier `address`, qui jouera le rôle d'adresse sur le réseau simulé.

Ajoutez un constructeur qui reçoit le nom en paramètre et définit l'adresse à 0 initialement.

Ajoutez un accesseur (*getter*) pour le nom et l'adresse.

- 2 Dans le constructeur, levez une exception si le nom n'est pas une chaîne entre 2 et 20 caractères.

Écrivez deux tests unitaires pour vérifier la bonne levée de l'exception ci-dessus.

/2

/2



/2

- 3 Créez une classe `ModemRouter` (qui représente la *box* de votre réseau, c'est-à-dire le point d'accès vers internet).

Cette classe possède les attributs suivants :

- Une liste `connected` de `Computer` connectés, initialement vide.
- Un entier `last_id`, qui représente la dernière adresse (le dernier identifiant) attribué. Initialement cet entier vaut 0.

/3

- 4 Ajoutez une méthode `int connect(Computer c)` qui permet de connecter un ordinateur. La méthode ajoute l'ordinateur à la liste, incrémente l'entier `address` puis le retourne.

Affichez également un message sur la console indiquant :

```
L'ordinateur <nom de l'ordi> est connecté avec l'adresse <address>
```

/2

- 5 Dans la classe `Computer`, ajoutez une méthode `void connectTo(ModemRouter mr)`. Cette méthode doit utiliser la méthode `connect` précédente pour connecter la machine courante au routeur, et définir l'attribut `address`.

/1

- 6 Écrivez le test unitaire suivant pour la méthode `connectTo` et assurez vous qu'il passe :

```
@Test
void test_connectTo() {
    Computer c = new Computer("bazinga");
    ModemRouter mr = new ModemRouter();
    c.connectTo(mr);
    int expected = 1;
    int actual = c.getAddress();
    assertEquals(expected, actual);
}
```

/2

- 7 Ajoutez un attribut `ModemRouter connectedTo` dans la classe `Computer`. Ajustez la méthode `connectTo` pour enregistrer cette information au moment de la connexion.

/1

- 8 Dans `Computer` écrivez une méthode `void receiveMsg(Computer from, String text)` qui affiche simplement le texte suivant sur la console :

```
<nom_du_computer> a reçu un message de <nom_de_from> : <text>
```

/2

- 9 Dans `ModemRouter`, écrivez une méthode `void transmit(Computer from, int dest, String text)` qui envoie un message à l'ordinateur spécifié par `dest`. Plus précisément cette méthode doit chercher dans la liste des `Computer` connectés celui qui possède l'adresse `dest`, puis appeler la méthode `receiveMsg` de ce `Computer`.

/1

- 10 Dans `Computer`, écrivez maintenant une méthode `void sendMsg(int dest, String text)` qui appelle la méthode `transmit` écrite précédemment.

Voici un code que vous pouvez tester dans une méthode `main` d'une classe `App` que vous créez :

```
Computer c1 = new Computer("nri");
Computer c2 = new Computer("irn");
ModemRouter mr = new ModemRouter();
c1.connectTo(mr);
c2.connectTo(mr);

c1.sendMsg(c2.getAddress(), "Salut !");
c2.sendMsg(c1.getAddress(), "Salut ! bien reçu.");
```

Le résultat doit être :

```
L'ordinateur nri est connecté avec l'adresse 1
L'ordinateur irn est connecté avec l'adresse 2
irn a reçu de nri : Salut !
nri a reçu de irn : Salut ! bien reçu.
```

- 11 Dans ModemRouter, écrivez une méthode `List<Integer> getConnectedAddresses()` qui retourne la liste des adresses de tous les ordinateurs connectés. /2

Dans Computer, écrivez une méthode `void randomSendMsg(String text)` qui envoie un message à l'un des autres ordinateurs, choisi aléatoirement, connectés au ModemRouter.