



## TP n° 3: Classe et héritage (première partie)

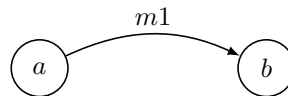
### Transmission de message entre les sommets d'un graphe

#### 0.1 Exercice 1 : Découverte de l'outil Socket

1. En utilisant la documentation officielle, étudier les classes `tp3.Programme1` et `tp3.Programme2`.
2. Dans un terminal compiler puis exécuter la classe `tp3.Programme1`, et dans un autre terminal compiler et exécuter la classe `tp3.Programme2`. Que font ces programmes ?
3. Apporter les modifications suivantes en expliquant les résultats et/ou erreurs produits :
  - (a) Modifier la valeur de `portDestination` de la classe `tp3.Programme2`
  - (b) Donner les mêmes valeurs à `portDestination` et à `port` de la classe `tp3.Programme2`
  - (c) Ajouter une classe `tp3.Programme3` qui envoie un autre message à `tp3.Programme1`, puis exécuter dans l'ordre les classes `tp3.Programme1`, `tp3.Programme2` et `tp3.Programme3` dans des Terminal différents.
4. Remettre les classes `tp3.Programme1` et `tp3.Programme2` à leur état initial, puis modifier `tp3.Programme1` de manière à résoudre la question 3c
5. Modifier `tp3.Programme1` et `tp3.Programme1` de manière à pouvoir envoyer un objet de type `tp3.Message`. Un objet `tp3.Message` doit contenir le numéro de port source, le numéro de port destination et le contenu du message envoyé. Celui-ci doit être saisi par l'utilisateur au moment de l'envoi et affiché à la réception. Les numéros de port sont des valeurs fixes.

#### 0.2 Exercice 2 : Sommet émetteur et sommet récepteur

Dans cette partie nous voudrions modéliser la communication entre deux sommets d'un graphe : un émetteur et un récepteur (voir la figure suivante)



- Un émetteur et un récepteur sont tous les deux sommet d'un graphe. L'émetteur est le sommet **a** et le récepteur est le sommet **b**
- L'émetteur et le récepteur contiennent chacun : un numéro de port et une adresse initialisée par "localhost"
- L'émetteur doit envoyer un objet de type **Message** au récepteur vers l'adresse et le port de destination
- Le récepteur reste à l'écoute sur son numéro de port pour recevoir l'objet **Message**.
- À la réception de l'objet **Message**, le récepteur doit afficher le contenu reçu.

Créer les classes appropriées dans le package `tp3` en utilisant les classes développées dans le deuxième TP.

#### 0.3 Exercice 3 : Synthèse

1. Pour pouvoir lancer cette simulation, il a fallu exécuter le programme émetteur et le programme récepteur dans deux JVM séparées. Peut-on regrouper la simulation dans un seul environnement ? Pourquoi ?
2. Quelles sont les différences entre les deux classes de émetteur et récepteur ?
3. Avec la modèle utilisé dans ces classes, peut-on modéliser le graphe suivant (où **a** est à la fois émetteur et récepteur) ? Quelles sont les limites ?

