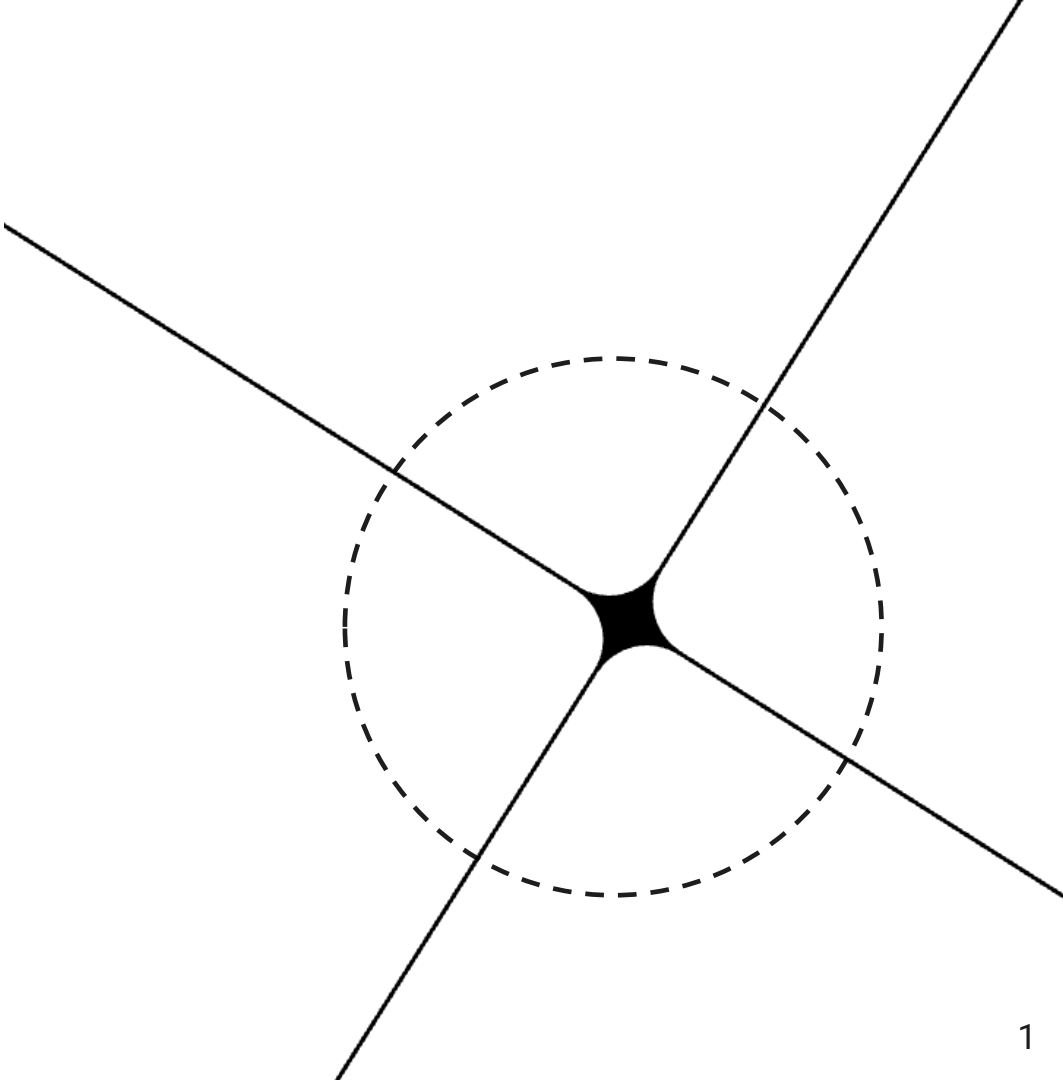
INSTITUT POLYTECHNIQUE DE PARIS

TELECOM SudParis

Introduction TP 7

Julien Romero



Dans ce TP…

- Nous allons créer un site web pour afficher des films et des avis
  - Gestion d'une base de données simple avec les collections de Java
  - Création d'un serveur web capable de servir un site web
- Ce TP est purement pédagogique, les sites web sont généralement créés avec des frameworks, c'est-à-dire un ensemble de composants logiciels accélérant la création d'une application
  - Spring en Java, Django en Python, Node.js et React en Javascript
- Pour des cas particuliers ou des optimisations, on peut avoir à le réimplémenter un serveur nous-même
  - Transmission de données brutes, optimisation des communications, protocole personnalisé, …

Retour sur toString()

- Tous les objets en Java héritent de `public String toString()` car cette méthode est dans Object
  - L'implémentation par défaut est basée sur une adresse mémoire de l'objet et ne fait pas de sens
- Quand on redéfinit `public String toString()`, on doit retourner une version textuelle de l'objet actuel
  - On n'appelle jamais `System.out.println(...)` dans toString() !
- La plupart du temps, `public String toString()` est appelée implicitement sans avoir besoin de le faire vous même
  - `System.out.println(myObject); System.out.println("Value: " + myObject);`
  - `String s = ""; s += myObject;`
  - Certains frameworks pour écrire des logs font aussi les conversions implicitement
- Quand appeler `toString()` ?
  - Quand on appelle une méthode prenant un `String` en entrée
  - Quand on veut construire une `String` sans utiliser une concaténation

Qu’est-ce qu’un protocole en informatique ?

- Un protocole est un ensemble de règles définissant comment des entités (machines, applications) communiquent
  - On spécifie le format des messages, l’ordre des informations, la gestion des erreurs, etc.
- Les protocoles permettent l’interopérabilité entre des systèmes différents
  - En utilisant le même protocole, tout le monde parle la même langue et peut communiquer
- Les protocoles peuvent être
  - Standardisés: des organismes fixent les règles des protocoles. C’est le cas des protocoles utilisés sur le web
  - Personnalisés: vous pouvez vous-même créer votre protocole
    - Ex: pour communiquer une information sur un personnage, je pourrais utiliser 4 bits pour dire que je parle d’un personnage, 5 bits pour donner son identifiant, 3 bits pour ses points de vie, 1 bit s’il possède une épée, … À la fin, on obtient une suite de bits comme 0111010001001000, qui n’est compréhensible qu’avec les règles du protocole

TCP/IP: le cœur de la communication réseau

- TCP/IP est un ensemble de protocoles qui sous-tendent l'internet
- Découpage des protocoles en plusieurs couches, en partant de couches basses (physique) aux couches hautes (applications). Voici les couches avec des exemples de protocoles :
  - Physique : conventions sur les caractéristiques physiques des communications (cables, fibre, radio, codage, modulation, …)
  - Liaison : Ethernet, Wi-Fi, Bluetooth, Zigbee
  - Réseau : IP
  - Transport : TCP (transport fiable, sans perte), UDP (transport non fiable, avec perte, en général pour du multimédia)
  - Application : HTTP, FTP, DNS, SSH
- Voir les cours de réseau pour plus de détails

L'adresse IP et les ports

- Une adresse IP est une suite de nombre permettant d'identifier une machine sur un réseau
  - Comme une adresse postale
  - IPv4 : 157.159.11.11 (machine contenant le site web telecom-sudparis.fr)
  - IPv6 : 1050:0000:0000:0000:0005:0600:300c:326b (plus d'adresses)
- Un port est un numéro permettant d'identifier un service sur une machine
  - En général, 80 correspond à un site web en HTTP, 443 à un site web en HTTPS, 22 pour SSH
  - 8080 souvent utilisé pour du développement en local
  - L'adresse complète du site de Télécom Sudparis est 157.159.11.11:80 (adresse IP:port)
- Une communication TCP ou UDP est identifiée par une adresse IP source, un port source, une adresse IP destination, et un port destination

Les communications réseaux en Java

- Java implémente des sockets pour la communication réseaux
  - Une socket est une bibliothèque permettant de facilement écrire sur la couche transport (TCP, UDP)
  - Il n'y a pas encore la couche application, il faut la faire vous-même !
- Pour utiliser une socket, il faudra d'abord établir une connexion, puis faire la communication à proprement parler, et enfin fermer la connexion
- Par défaut, une socket nous donne des fonctions pour lire et écrire des bytes (InputStream/OutputStream)
  - Il existe des méthodes facilitant la lecture et écrire de texte (BufferedReader, PrintStream)
- Voir le TP pour plus de détails

HTTP : protocole de l'Hypertexte / du Web

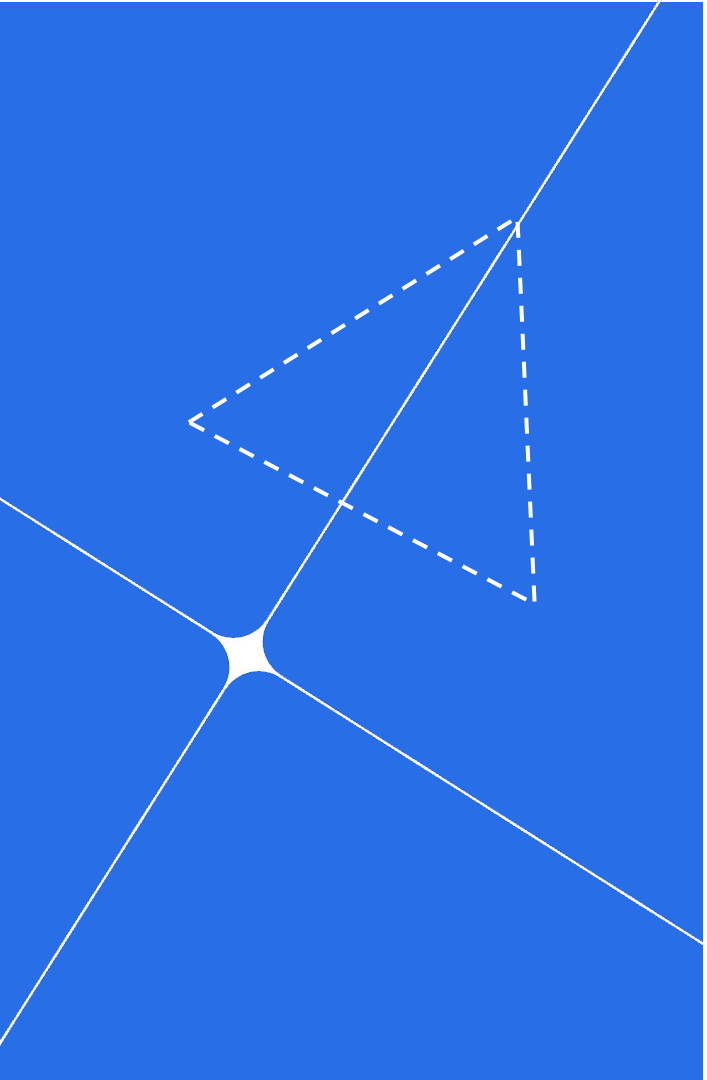
- HTTP est un protocole de la couche applicative de TCP/IP
- Permet au client (navigateur ou programme) de demander des ressources (pages HTML, images, données…) à un serveur Web, et au serveur de répondre.
    - Protocole lisible par un human (vraies chaînes de caractères, pas juste des bits)
- Dans le TP, nous allons implémenter une partie du protocole HTTP permettant à votre navigateur de demander un site web au serveur

HTML : langage de structuration des pages Web

- HTML est un langage de balisage permettant de structurer le contenu d'une page Web : titres, paragraphes, listes, liens, etc.
- Une balise délimitant du contenu a le format <balise paramètres>contenu</balise>
  - Les balises sont imbriquées les unes dans les autres
- Exemples de balises :
  - <p> : paragraphe
  - Texte : lien vers *url* en cliquant sur *Texte*
  - : liste
  - : élément d'une liste
- Exemple:

```
<ul>
  <li>premier élément de la liste</li>
  <li>second élément de la liste</li>
</ul>
```

- Votre serveur web utilisera HTTP pour envoyer une page en HTML
- Toutes les pages web sont in fine interprétée en HTML (+CSS pour le style, + Javascript pour la dynamicité)



En route pour le TP