

TD n°1 : Introduction au diagramme de classes & premiers pas en java

Exercice 1 : Un petit échauffement sur le diagramme de classes ...

Pour chaque question suivante, proposez un diagramme de classes correspondant à la description donnée :

1. Une **personne** possède un ou plusieurs **ordinateurs**.
2. Considérons les **processus** qui s'exécutent dans un ordinateur.
La liste des processus faisant appel à une ressource est gérée par une **file d'attente** (comme dans le cas d'une imprimante par exemple). Cette file stocke les processus en attente de la ressource.
Chaque processus a bien sûr un **numéro** unique dans la file et ne peut appartenir qu'à une seule file d'attente. Proposez une modélisation pour la file d'attente.
3. Un **fichier** est accessible par un **utilisateur** selon des **droits d'accès** en **lecture** et en **écriture**.
4. Un message électronique (**e-mail**) a un **titre**. Un e-mail est toujours composé d'un **corps** et d'un **en-tête**. Le corps d'un e-mail a un **message** et une **signature**. Des **pièces jointes** peuvent éventuellement être ajoutées à un e-mail. Une même pièce jointe peut être partagée par plusieurs e-mail.
Un e-mail contient également 1 ou plusieurs **destinataires** que l'on peut sélectionner soit par leur **nom**, soit par leur **adresse électronique**.
Dans votre modélisation, utilisez à bon escient des relations de composition et d'agrégation...

Remarques :

- Nous considérerons que le **corps d'un e-mail** a un message et une signature.
- Quant à **l'en-tête d'un e-mail**, elle permet de connaître certaines données concernant le "parcours" du message sur Internet. Les données contenues dans une en-tête d'e-mail ne nous intéressent pas pour l'instant...Vous pourrez toujours enrichir plus tard votre modélisation en visualisant sur votre messagerie l'entête d'un de vos e-mail...

Exercice 2 : Les débuts de l'informatique moderne :

Charles Babbage, Ada Lovelace, Alan Turing & Co...

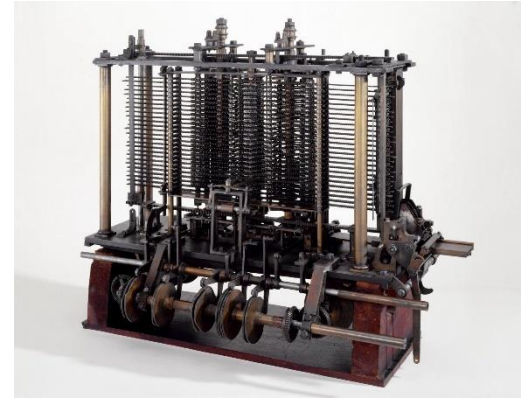
Vous avez choisi de faire des études dans le domaine de l'Informatique (Computer Science).
Mais connaissez-vous l'histoire de l'informatique ?

L'histoire de l'informatique a commencé bien avant la discipline moderne des sciences informatiques, généralement par les mathématiques ou la physique. Les développements des siècles précédents ont évolué vers la discipline que nous connaissons aujourd'hui sous le nom d'informatique. Cette progression, des inventions mécaniques et des théories mathématiques vers les concepts et les machines informatiques modernes, a conduit au développement d'un domaine académique majeur, à un progrès technologique spectaculaire à travers le monde occidental et à la base d'un commerce et d'une culture mondiale massive (extrait https://fr.wikipedia.org/wiki/Histoire_de_l%27informatique)



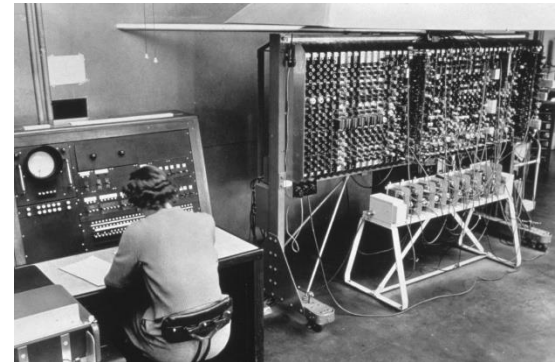
Ada Lovelace, de son nom complet Augusta Ada King, comtesse de Lovelace, née Ada Byron est une pionnière de la science informatique.

Elle est principalement connue pour avoir réalisé **le premier véritable programme informatique**, lors de son travail sur un ancêtre de l'ordinateur : la **machine analytique de Charles Babbage**.



Alan Mathison Turing est un mathématicien et cryptologue britannique, auteur de travaux qui fondent scientifiquement l'informatique.

Pour résoudre le problème fondamental de la décidabilité en arithmétiques, il présente en 1936 une expérience de pensée que l'on nommera ensuite **machine de Turing** et des concepts de programme et de programmation, qui prendront tout leur sens avec la diffusion des ordinateurs, dans la seconde moitié du xx^e siècle.



Textes et images extraits de :

https://fr.wikipedia.org/wiki/Ada_Lovelace , https://fr.wikipedia.org/wiki/Alan_Turing , <https://www.sciencemuseum.org.uk/objects-and-stories/lovelace-turing-and-invention-computers>

Votre cahier des charges : Imaginez maintenant que le département Informatique soit votre nouveau client et que pour les JPO, il vous demande de créer un jeu (une sorte de memory par exemple) qui permettrait d'associer à quelques personnages célèbres de l'histoire de l'informatique la machine sur laquelle ils ont travaillé.

Mettre en place ce jeu nécessite donc de pouvoir manipuler des personnages célèbres (**famous computer pioneer**) et des machines (**device**).

❑ Focus sur la classe Device :

L'application devra manipuler des objets de type **Device**, tels que la machine de Babbage ou la machine de Turing, ...

Pour l'instant, seuls le nom et l'année d'invention de cette machine nous intéressent.

1. Après cette rapide **analyse**, vous pouvez passer à la **phase de conception**.
Commencez par modéliser la classe **Device** à l'aide d'un diagramme UML en tenant compte des besoins du client. Vous modéliserez cette classe dans la rubrique **Conception** du tableau récapitulatif à compléter en fin de cet énoncé (voir page 4).
Quels sont ses attributs, ses méthodes ?
A-t-on besoin d'ajouter des setteurs et/ou des getteurs dans le contexte du client ?
Que pouvez-vous dire sur le(s) constructeur(s) à écrire ?
Justifiez vos réponses.
2. Après la conception vient la **phase d'implémentation** et votre langage de prédilection est désormais le Java !
Implémentez la classe **Device** en Java dans la rubrique **Implémentation** du tableau récapitulatif à compléter en fin de cet énoncé (voir page 4).

3. Après la phase d'implémentation vient la **phase de test**.

Pour tester votre implémentation, vous complétez la rubrique **Tests** du tableau récapitulatif donné en fin de cet énoncé (voir page 4), en implémentant une classe **Main** avec une méthode **main** de signature `public static void main(String[] args)` qui contiendra le premier jeu d'essai suivant :

- **création d'une instance** (objet) de type **Device**.
Cet objet sera nommé `babbageMachine` et instancié avec les *bonnes* valeurs qui permettront de procéder à l'affichage suivant sur la console :
The Babbage Analytical Machine was invented in 1837.
- **l'affichage** sera obtenu en utilisant l'instruction `System.out.println` qui prendra juste en paramètre un appel à la méthode `toString()` (qui comme son nom l'indique renvoie un **String**) sur l'objet `babbageMachine`.
La méthode `toString()` devra bien sûr être ajoutée à la fois dans votre diagramme UML et dans votre implémentation Java ! C'est une méthode qui, comme ici, est habituellement utilisée pour renvoyer l'état d'un objet (rappel : l'état d'un objet correspond à la valeur de ces attributs).

Si le temps vous le permet, vous pourrez également implémenter un second jeu d'essai qui procédera cette fois-ci à :

- **Instanciation** d'un objet de **Device** nommé `turingEngine` dont l'appel de la méthode `toString()` sur cet objet permettra de procéder via un `System.out.println` à l'affichage console suivant :
The Turing Engine was invented in 1936.

Conception <i>Diagramme de classes (UML)</i>	Implémentation <i>Ecriture du code dans le langage Java</i>
Tests (manuels)	



Travail à faire pour la semaine prochaine : **pour réviser ce TD et préparer les TD et TP de la semaine** **prochaine 😊**

Dans cette ressource nous utiliserons parfois une pédagogie traditionnelle, parfois une pédagogie inversée, disons que ce sera de la **pédagogie semi-inversée** 😊 que ce soit à la fois pour les cours, les TD et les TP...



Pour la séance de TD de la semaine prochaine, vous devez regarder les vidéos suivantes qui ont été réalisées par José Paumard, enseignant-chercheur à Paris 13, mais aussi Java Champion et Java DevRel chez Oracle 😊 Vous pouvez suivre José Paumard sur <https://twitter.com/JosePaumard>

→ Dans la rubrique **Ecrire une première classe en Java**

Accès direct à la rubrique via : <https://unil.im/butoo1>

(URL complète : https://www.youtube.com/playlist?list=PLzzeuFUy_CniXWmmdo8zmgo3bXdqy4aG8)

... Dans cette rubrique, vous vous focaliserez pour le moment sur les vidéos suivantes ...
(environ 20 minutes)

❑ **01. Création d'une première classe User et première instanciation (6:31)**

⇒ Accès direct à la vidéo via : <https://unil.im/butoo11>

❑ **02. Organisation d'une instance de User en mémoire (3:09)**

⇒ Accès direct à la vidéo via : <https://unil.im/butoo12>

❑ **03. Invocation d'une méthode de User (4:28)**

⇒ Accès direct à la vidéo via : <https://unil.im/butoo13>

❑ **05. Classe, type référence, objet et instance en programmation objet (7:28)**

⇒ Accès direct à la vidéo via : <https://unil.im/butoo14>

Facultatif pour le moment (plus facile à comprendre si vous connaissez le langage C 😊) :

❑ 04 Réserve et libération de la mémoire par le garbage collector

C'est par là pour voir la suite des vidéos à regarder 😊

→ Dans la rubrique **Structure d'une classe**

Accès direct à la rubrique via : <https://unil.im/butoo2>

(URL complète : https://www.youtube.com/playlist?list=PLzzeuFUy_Cnh_jAwFXkYMjRd9wYhObqAL)

... Dans cette rubrique, vous vous focaliserez pour le moment sur les vidéos suivantes ...
(environ 15 minutes)

❑ **01. Champs, constructeur, méthode et mot clé `this` (4:07)**

⇒ Accès direct à la vidéo via : <https://unil.im/butoo21>

❑ **02. Modificateur de visibilité : `public` et `private` (4:27)**

⇒ Accès direct à la vidéo via : <https://unil.im/butoo22>

❑ **10. Mot clé `final` sur une classe, une méthode ou un champ (6:21)**

⇒ Accès direct à la vidéo via : <https://unil.im/butoo23>

→ Pour préparer le prochaine TD : dans la rubrique **Classes Object et String**

Accès direct à la rubrique via : <https://unil.im/butoo3>

(URL complète : https://www.youtube.com/playlist?list=PLzzeuFUy_CnhW4RoeaQ36pZ5tqoK5lxr7)

... Dans cette rubrique, vous vous focaliserez pour le moment sur les vidéos suivantes ...
(environ 30 minutes)

❑ **01. Toutes les classes Java étendent la classe `Object` (4:57)**

⇒ Accès direct à la vidéo via : <https://unil.im/butoo31>

❑ **02. Méthode `equals()` de la classe `Object` (6:17)**

⇒ Accès direct à la vidéo via : <https://unil.im/butoo32>

❑ **03. Redéfinition de `equals()` par surcharge (7:45)**

⇒ Accès direct à la vidéo via : <https://unil.im/butoo33>

❑ **04. Surcharge de `equals` dans la classe `String` (3 :48)**

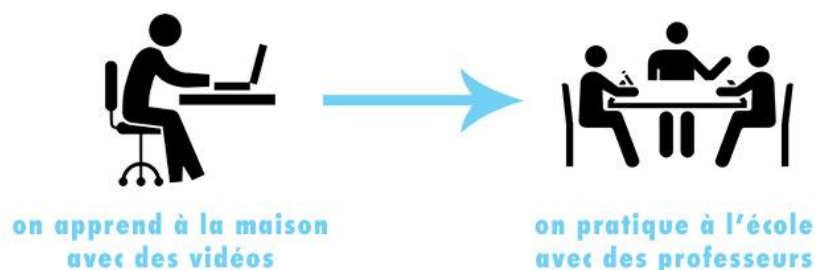
⇒ Accès direct à la vidéo via : <https://unil.im/butoo34>

❑ **06. Surcharge de la méthode `toString()` (5 :15)**

⇒ Accès direct à la vidéo via : <https://unil.im/butoo35>

Facultatif pour le moment, mais comme les méthodes `equals` et `hashCode` seront directement générés par l'IDE en TP, visionner ces vidéos vous permettra de mieux comprendre au moment venu pourquoi les deux sont générés en même temps 😊

❑ **05. Méthode `hashCode()` liée à la méthode `equals`**



Images extraites de : <https://www.learningbydoing.fr/pedagogie-inversee> et <https://www.videotelling.fr/la-classe-inversee>