# Chapitre 1 - Introduction Java:

## Intro :

Java a été créée par Sun Microsystems en 1995. Il est encore très populaire

Sur Java il compile en bytecode, sur Android c’est différent. (plus efficace).

Il est un langage orienté objet

## Structure d’un programme Java

Un programme est composé d’une ou plusieurs classes  
Une classe contient une ou plusieurs méthodes  
Une méthode contient des instructions  
Une Application Java contient généralement une méthode appelée main  
Voir lincoln.java

### Commentaires

Les commentaires dans un programme sont appelés documentation interne, ils expliquent le but du programme et décrivent les étapes de traitement mais n’influencent pas l’exécution du programme.

Ils peuvent prendre 3 formes :

// commentaire qui s’étend sur le reste de la ligne

/\* commentaire qui peut s’étendre sur plusieurs lignes \*/

/\*\* commentaire pour javadoc

### Variables

Les identificateurs sont des mots utilisés par le programmeur, composés de lettres, de chiffres, tiret bas et signe de dollar. Ils ne peuvent commencer par un chiffre

Java fait la différence entre majuscules et minuscules

Par convention :

Le nom des classes débutent par une majuscule

Le nom d’une constante est tout en majuscule

### Mots réservés

|  |  |  |  |
| --- | --- | --- | --- |
| abstract  boolean  break  byte  case  catch  char  class  const  continue  default  do  double | else  extends  false  final  finally  float  for  goto  if  implements  import  instanceof  int | interface  long  native  new  null  package  private  protected  public  return  static  strictfp | super  switch  synchronized  this  throw  throws  transient  true  try  void  volatile  while |

### Séparateurs

Séparateur : espace, ligne vide, tabulation

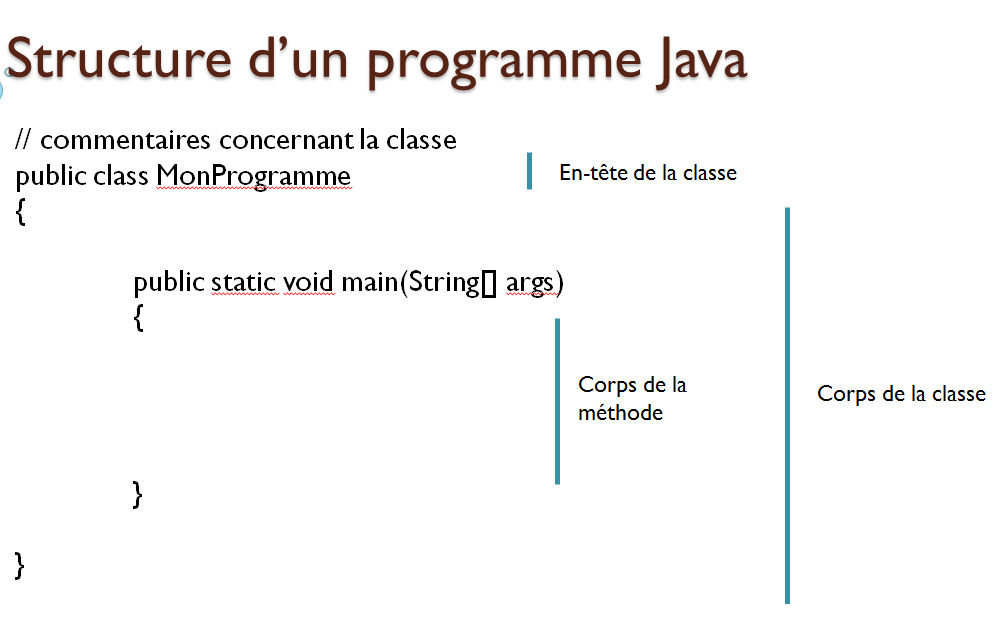
Un séparateur est utilisé pour séparer des mots et des symboles

Les séparateurs en trop sont ignorés

Un programme peut être formaté de différentes façons

Les programmes doivent être formatés de façon à augmenter la lisibilité en utilisant une indentation adéquate

Voir Lincoln2 et Lincoln3



## Niveaux des langages

Le compilateur Java traduit le code source en bytecode  
Le bytecode n’est pas un langage machine  
Un interpréteur traduit le bytecode en langage machine et l’exécute  
Donc, un compilateur Java n’est pas lié à une architecture CPU particulière  
On dit donc que Java est PORTABLE

## Environnement de développement

Il y a plusieurs environnements pour développer des logiciels Java :

Eclipse

jGRASP

Sun Java Development kit (JDK)

Sun Forte for Java

Monash BlueJ

Microsoft Visual J++

Netbeans

## Syntaxe et sémantique :

Les règles de syntaxe définissent comment nous pouvons combiner des symboles, des mots réservés et des identificateurs pour faire un programme valide

La sémantique définit la signification des instructions

Autrement dit, un programme peut être correct au niveau de la syntaxe sans l’être au niveau du sens (sémantique)

Un programme fait ce qu’on lui dit de faire et non ce qu’on pensait lui dire de faire

### On voit 3 types d’erreurs :

Erreurs de compilation : le compilateur les trouve et les signale. L’exécutable n’est pas généré

Erreur d’exécution

Erreur de logique

Erreur d’allocation (new to create, garbage collector to delete) Le garbage collector

Erreur de dépassement

# Chapitre 2 - Données et expressions :

## Chaines de caractères

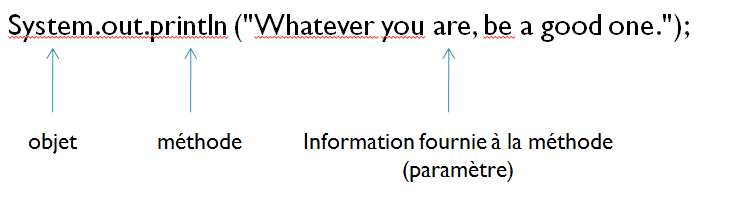
Chaque chaîne de caractères est un objet en Java, défini la classe String

Chaque chaîne littérale (entre guillements anglais) représente un objet String

## Utilisation d’objets

L’objet System.out représente une destination vers la quelle nous pouvons envoyer des sorties

Dans Lincoln, on invoque la méthode println pour l’objet System.out



## La méthode print

L’objet System.out offre un autre service

print ressemble à println mais elle ne fait pas de saut de ligne

Donc, tous les affichages avec print apparaissent sur la même ligne

Voir Countdown.java

## Chaîne de caractères

L’opération de concaténation (+) permet de coller une chaîne à une autre

Une chaîne ne peut être divisée entre 2 lignes dans un programme

Voir Facts.java

L’opérateur + est aussi utilisé pour l’addition

La fonction de cet opérateur dépend du contexte

L’opérateur + est évalué de gauche à droite

Cet ordre peut être modifié à l’aide de parenthèses

Voir Addition.java

## Séquences Échappe

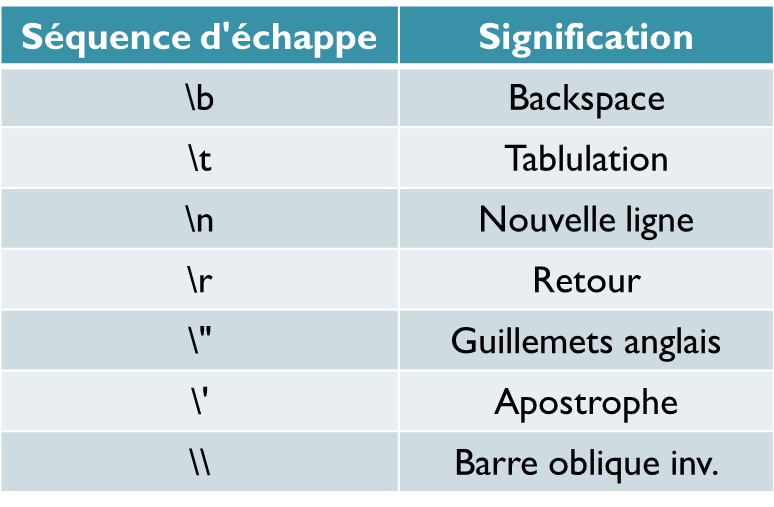
L’instruction ci-dessous est ambiguë pour le compilateur :

System.out.println("il a dit " bonjour" aux gens");

Une séquence Échappe est une série de caractères qui représente un caractère spécial

Une telle séquence commence par une barre oblique inversée indiquant que le caractère qui suit est spécial

System.out.println("Il a dit \"bonjour\" aux gens");



Voir Roses.java

## Variables

Une variable est un nom pour une zone en mémoire  
  
Une variable doit être déclarée en spécifiant :  
 son nom  
 le type de donnée qu'elle contiendra

Une variable peut recevoir une valeur initiale dans la déclaration  
 int sum = 0;  
 int min = 7, max = 149;

Quand une variable est référencée dans un programme, sa valeur courante est utilisée

Voir PianoKeys.java

## Constantes

Une constante est un identificateur qui ressemble à une variable sauf qu'elle garde UNE valeur tant que le programme est actif

Si on tente de modifier sa valeur, il y aura une erreur de compilation

En Java, on utilise final pour déclarer une constante

Constantes :

Nomment des valeurs littérales

Facilitent la modification de valeurs utilisées plusieurs fois dans le programme

Permettent d'éviter des modifications accidentelles de valeurs

## Types primitifs

Il y a exactement 8 types primitifs en Java

Quatre représentent les entiers :

byte, short, int, long

Deux représentent les réels :

float, double

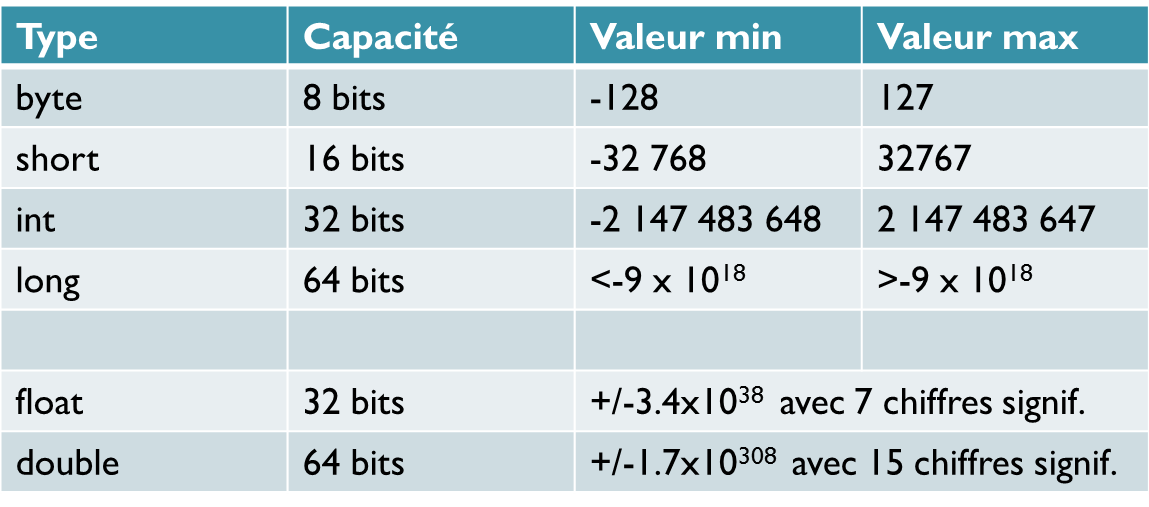
Un représente les caractères :

char

Un représente les booléens :

Boolean

Ce qui différencie les types primitifs numériques est leur taille :



## Caractères

Une variable char contient un caractère de l'ensemble Unicode

Chaque caractère correspond à un nombre unique

L'ensemble Unicode utilise 16 bits, donc il contient 65 536 caractères différents

C'est un ensemble international

Un caractère littéral est mis entre apostrophes

L'ensemble de caractères ASCII est plus vieux et plus petit qu'Unicode, cependant, il est encore populaire.

Les caractères ASCII sont un sous-ensemble de l'ensemble Unicode, incluant :

lettres majuscules et minuscules

ponctuation

chiffres

symboles spéciaux

caractères de contrôle (retour, tabulation…)

## Booléens

Une valeur booléenne est : vrai ou fausse

Exemple : ampoule allumée ou éteinte

Exemple : boolean fini = false;

## Expressions arithmétiques

Une expression est une combinaison d'un ou plusieurs opérandes et leurs opérateurs

Une expression arithmétique fait un calcul numérique en utilisant : +, -, \*, /, %

Si l'un des opérandes est réel, le résultat sera réel

Les opérateurs peuvent être combinés dans des expressions complexes

resultat = total + compteur / max - offset

Les opérateurs ont un ordre de priorité précis

\*, /, % sont plus prioritaires que + et –

Les opérateurs ayant la même priorité sont évalués de gauche à droite

Les parenthèses permettent de changer les priorités

L'opérateur d'affectation a une priorité plus basse que les opérateurs arithmétiques  
 resultat = somme / 4 + MAX \* petit;

Les côtés gauche et droit de l'affectation peuvent contenir la même variable  
 compteur = compteur + 1;

Ces opérateurs sont arithmétiques et unaires

++ incrémente de 1

-- décrémente de 1

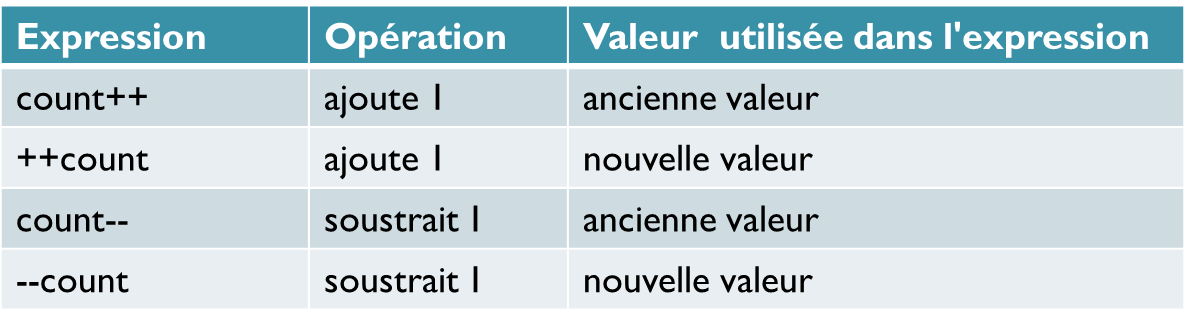
compteur++; est équivalent à :

compteur = compteur + 1;

Il y a 2 formes pour ++ et –

Postfixe : compteur++; ou compteur--;

Préfixe : ++compteur; ou --compteur;

Quand les opérateurs ++ et -- sont utilisés dans une expression plus grande, les 2 formes ont des effets différents 

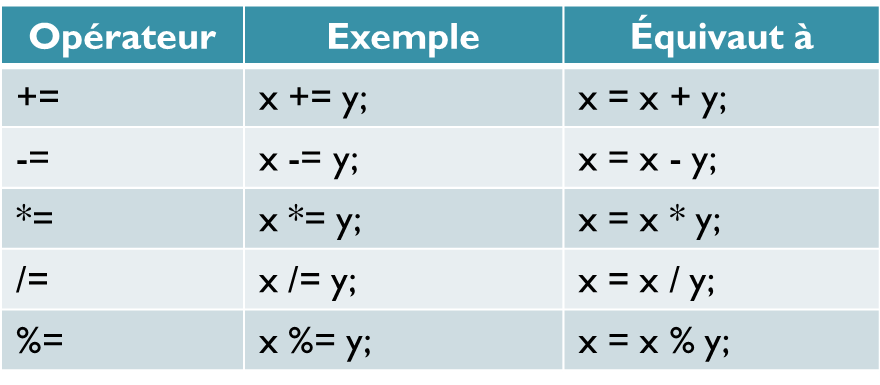
Souvent nous effectuons une opération sur une variable et nous stockons ensuite le résultat dans la variable

Java offre des opérateurs d'assignation qui simplifient ce processus

Par exemple, les instructions suivantes donnent le même résultat   
 nombre += compteur;

nombre = nombre + compteur;

Il y a plusieurs opérateurs d'assignation.



Le côté droit de l'opérateur est entièrement évalué d'abord, ensuite le résultat est combiné avec la valeur initiale de la variable

Le comportement de ces opérateurs dépend du type des opérandes

Si les opérandes de += sont des chaînes de caractères, ce sera une concaténation

## Conversions

Parfois, il est pratique de convertir les données d'un type à un autre type  
 Exemple : entier vers réel

Les conversions peuvent faire perdre de l'information

Les conversions de type "promotion numérique" sont sécuritaires

Les conversions "dégradantes" peuvent faire perdre de l'information (ex : int vers short)

En Java, les conversions peuvent se faire de 3 façons :

affectation

promotion arithmétique

cast

Conversion par affectation pour promotion numérique seulement

La promotion arithmétique se fait automatiquement quand les opérateurs convertissent leurs opérandes

Le cast est puissant mais dangereux

Exemple : total et compteur étant des nombres entiers, on peut avoir une division réelle

resultat = (float) total/compteur

## Programmes interactifs

Généralement, les programmes ont besoin de lire des données

La classe Scanner fournit des méthodes pratiques pour lire des valeurs de différents types

Un objet Scanner peut lire des données à partir de différentes sources, incluant le clavier

Les entrées à partir du clavier sont représentées par l'objet System.in

La ligne suivante crée un objet Scanner qui lit à partir du clavier :

Scanner scan = new Scanner (System.in);

L'opérateur new crée l'objet Scanner

Une fois créé, l'objet Scanner peut être utilisé pour appeler différentes méthodes de lecture :

answer = scan.nextLine();

La classe Scanner fait partie de la librairie de classes java.util, et doit être importée dans un programme pour être utilisée

la méthode nextLine lit toute une ligne

Les détails concernant la création des objets sera présenté au chapitre 3

Voir Echo.java

Les séparateurs sont utilisés pour séparer le éléments (Tokens) lors de la lecture

Les séparateurs incluent les espaces, la tabulation et le saut de ligne

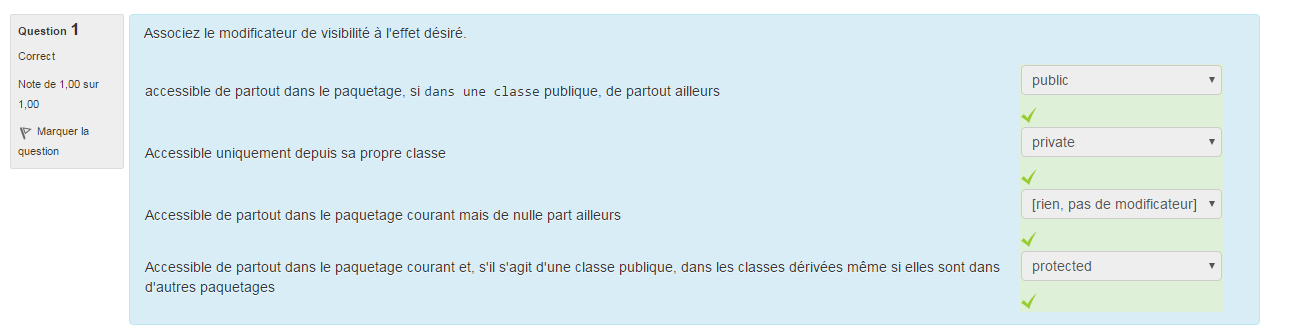
La méthode next de la classe Scanner lit le token suivant et le retourne dans une chaîne de caractères

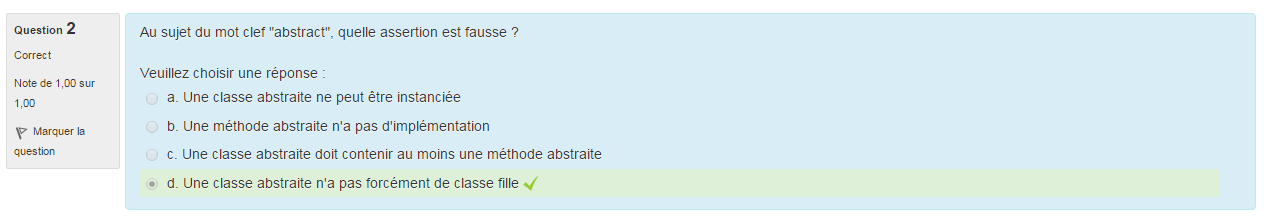
Des méthodes comme nextint et nextDouble lisent des données d'un type particulier

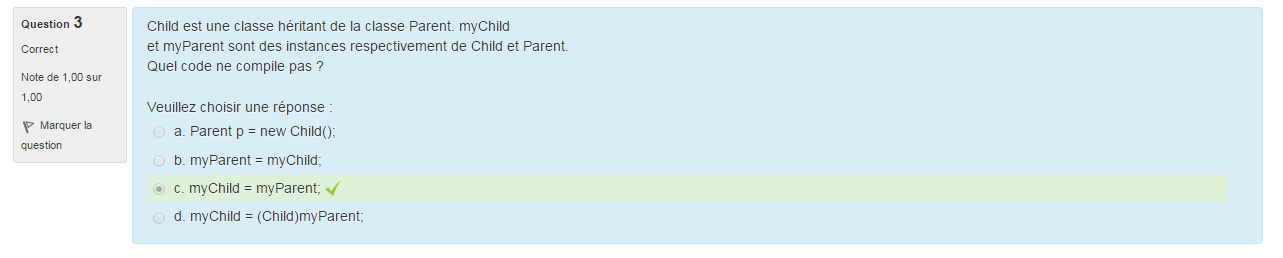
Voir GasMileage.java

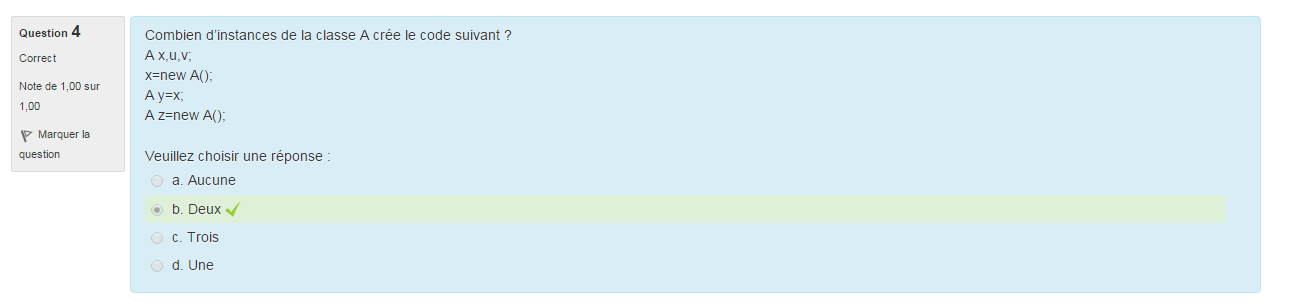
# Questionnaires :

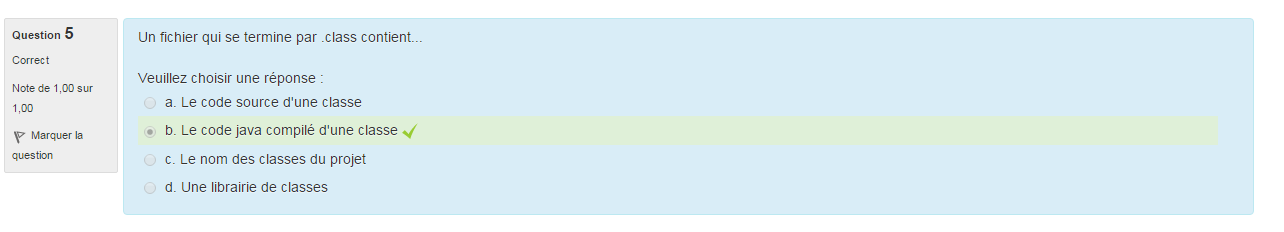
## Quiz 1 :











Quiz 2 :

1. Lequel des énoncés suivant est incorrect?

La réponse correcte est : Les classes et les objets sont juste des mots différents pour désigner la même chose

2. Lesquels parmi les types suivants peuvent être directement instanciés en Java :

La réponse correcte est : TreeMap, Vector, TreeSet, LinkedList, ArrayList, HashSet

3. Quelle sera le résulta à l'écran quand on exécute le code suivant?  
List<String> nombres = Arrays.asList(new String[]{"Un","Deux","Trois"});  
  
nombres.forEach(System.out::println);

La réponse correcte est : Un  
Deux  
Trois

4. Nous avons vu qu'en Java il existe deux zones de mémoire le "heap" et le "stack" (respectivement monceau et pile dans certains ouvrages en français). Par sa nature propre, un de ces espaces est sujet à se fragmenter, mais pas l'autre. Quel espace sera porté à se fragmenter?

Le "heap" est l'endroit ou sont alloué les objets**quand on utilise la commande "new"**. Ces objets sont détruits seulement s'il n'existe aucune référence vers eux. Le processus de "garbage collection" (ramasse-miettes en français) se charge de récupérer l'espace de ces objets inutilisés. Comme le système ne peut prédire dans quel ordre ces objets deviendront inutiles, le "heap" se comporte un peu comme un disque dur et accumule des objets et des trous qui apparaissent quand ces objets sont libérés. L'équivalent de la défragmentation s'appelle "compaction". Cette opération est prise en charge par le processus de "garbage collection".

Le "stack" est beaucoup plus prévisible car**il croit d'appel de méthode en appel de méthode** et **décroit à leur retour**. Il n'y a donc aucune fragmentation possible, donc un gain de performance si nos objets n'ont pas une longue durée de vie.

La réponse correcte est : Heap

5. public class Pair<K, V> {  
 public K getKey();  
 public V getValue();  
}

Que représentent K et V dans ce contexte?

Les "generics" sont apparus dans la 5ème version de Java. Ils permettent de définir des attributs et des méthodes dont les composantes ne sont pas prédéfinies. Ici K et V désignent les types qui seront utilisés pour l'ensembles des objets de classe "Pair" utilisant ces types. On peut donc définir, utiliser et composer des objets plus complexes à partir de composantes génériques. Le compilateur créera automatiquement les classes spécifiques au besoin.

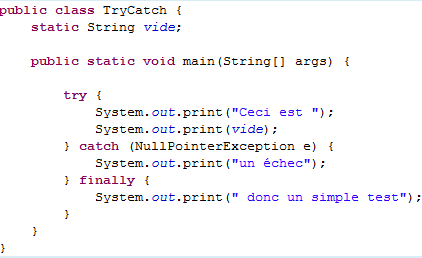
Pour instancier cette classe on devra donc spécifiet ces types au moment du "new" :

Par exemple:

Pair<String,Int> monObjet =  new Pair<String, Int>();

Dans cet exemple, je force monObjet à ne contenir que des pairs dont la clé est une String et la valeur un chiffre.

La réponse correcte est : Les types des objets servant de clé et valeur.

6. Qu'affichera le code suivant: 

La fonction "print" du System.out est capable d'afficher une valeur nulle sans erreur. Il n'y aura donc pas d'exception. Le bloc "finally" s'exécute dans tous les cas

La réponse correcte est : Ceci est null donc un simple test

7. Une librairie sous forme de fichier d'archive java porte quelle extension de fichier

La réponse correcte est : .jar

8. Une librairie sous forme de fichier d'archive java peut être ouvert avec Windows à condition qu'on en change l'extensions. Quelle est l'extension que Windows reconnaîtra?

La réponse correcte est : .zip

9. Quel mot clé est utilisé pour créer un objet à partir d'une classe?

La réponse correcte est : new

10. Que veut dire l'acronyme JIT

"Just In Time" réfère à la capacité de compiler dynamiquement le binaire Java (qui est par nature générique) en binaire spécifique à l'architecture de la machine physique sur lequel il est exécuté. Cette conversion se fait au besoin d'où l'appelation JIT qui signifie "Juste à temps".

La réponse correcte est : Just In Time

11. Une méthode d'instance ne peut pas être redéfinie en diminuant son niveau de visibilité.

La réponse correcte est « Vrai ».

12. En Java, un booléen peut avoir une valeur négative.

La primitive booléenne de Java est très stricte. Seulement deux valeurs sont possibles "True" ou "False".

La réponse correcte est « Faux ».

13. Le type "byte" en java contient une valeur entre 0 et 255.

Le type "byte" de Java est signé en complément de deux. Il a donc une étendue de -128 jusqu'à 127.

La réponse correcte est « Faux ».

14. Le JRE est forcément plus petit que le JDK

Le JDK inclut le JRE

La réponse correcte est « Vrai ».

15. La valeur par défaut d'un booléen est "faux"

La réponse correcte est « Vrai ».

16. En Java le type "char" est de 8 bits

Java supporte Unicode de façon native. Sa représentation interne est en 16 bits par caractères.

La réponse correcte est « Faux ».

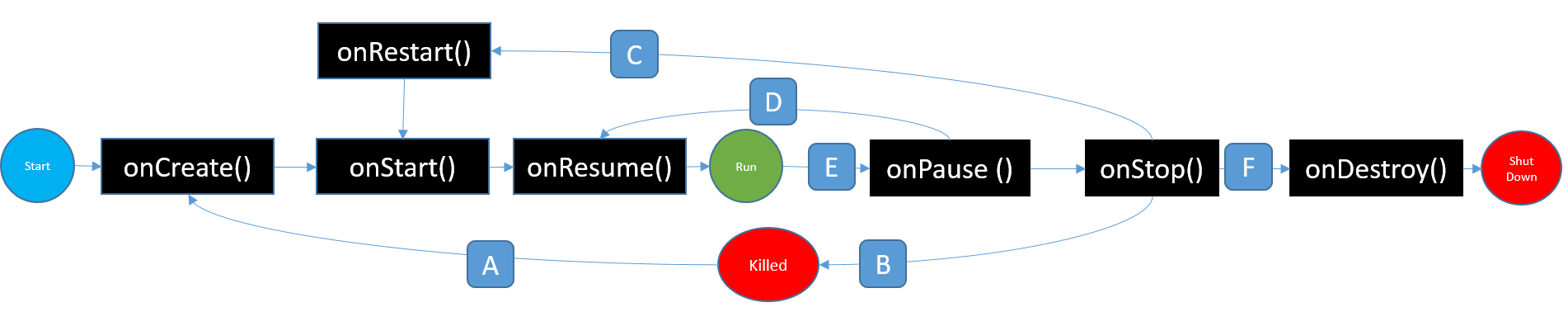
La quasi-totalité des attributs de la bibliothèque standard sont publics.

La quasi-totalité des attributs de la bibliothèque standard sont privés. Donner le statut privé à un attribut permet de garder un contrôle sur celui-ci et permet aussi de garder une cohérence entre les différents attributs.

La réponse correcte est « Faux ».

### Quiz 3 :

1.



La réponse correcte est :

A – L'utilisateur navigue vers l'activité.,

B – Une application plus prioritaire à besoin de mémoire.,

C – L'utilisateur navigue vers l'activité.,

D – L'utilisateur revient vers l'activité.,

E – Une autre activité vient à l'avant-plan.,

F – L'activité s'est terminé d'elle-même ou détruite par le système.

2. Associez la librairie Android à sa définition:

La réponse correcte est :

Riche collection de composantes d'interfaces graphiques. – android.widget,

Interface de programmation de rendu 3D. – android.opengl,

Pierre angulaire du modèle applicatif Android. – android.app,

Utilisé pour accéder au contenu publié par le fournisseurs de contenus, cette librairie inclut aussi un gestionnaire SQLite. – android.database,

Permet d'afficher et manipuler le chaines de caractères sur l'affichage. – android.text, Facilite l'accès aux contenus, leur publication et fournit la messagerie entre les applications anisi que leur composantes. – android.content,

Les éléments fondamentaux pour construire les interfaces usagers. – android.view,

Permet d'ajouter des fonctionnalités de fureteur internet dans une application. – android.webkit,

Fournit l'accès au messages du système, ses processus et à la communication inter-processus. – android.os

3. Les concepts suivants correspondent à des classes de l'environnement de développement Android. Associez les définitions aux bons termes.

La réponse correcte est :

permet d’écouter ce qui se passe sur le système ou sur votre application et déclencher une action que vous aurez prédéfinie.– Broadcast and Intent Receivers,

Unité de base des actions sous Android – Activity,

permet de gérer l'envoi et la réception de messages afin de faire coopérer les applications – Intent,

ne possède pas de vue mais permet l’exécution d’un algorithme sur un temps indéfini. Il ne s’arrêtera que lorsque la tâche est finie ou que son exécution est arrêtée. – Service,

permet de déléguer une action à un autre composant, une autre application ou une autre activité de l'application courante – Intent

4. Dans un fichier XML Android, le symbole @ dans une chaîne indique:

La réponse correcte est : Que le contenu est dans une autre ressource.

5. Quels énoncés sont vrais pour la méthode onCreate() d'une activité?

La réponse correcte est : 

Appelé quand l'activité est d'abord crée.,

C'est là où vous devez assigner les propriétés qui ne changent pas (vues, couplages des données, listes, …),

Cette méthode vous fournit aussi des données sur l'état précédent de l'activité mise au rancart.,

Toujours suivi de onStart()

6. Quels énoncés sont vrais pour la méthode onDestroy() d'une activité?

Vous en avez sélectionné correctement 2.

La réponse correcte est :

Dernière chance de faire quelque-chose avant l'extinction de l'activité.,

Appelé quand l'activité se termine.,

Appelé quand le système manque d'espace.

**7.** Quels énoncés sont vrais pour la méthode onPause() d'une activité?

La réponse correcte est : 

Appelé quand l'activité passe en arrière-plan.,

Appelé quand l'activité se termine.,

Appelé quand le système manque d'espace.

**8.** Quels énoncés sont vrais pour la méthode onRestart() d'une activité?

La réponse correcte est : 

Appelé quand une activité a été arrêté, avant de redémarrer.,

Toujours suivi de onStart()

**9.** Quels énoncés sont vrais pour la méthode onResume() d'une activité?

La réponse correcte est : 

Appelé quand l'activité commence les interactions avec l'utilisateur.,

À ce point votre activité est sur le dessus de la pile des activités. Les actions de l'usager y sont capturés.,

Toujours suivi de onPause()

**10.** Quels énoncés sont vrais pour la méthode onStart() d'une activité?

La réponse correcte est : 

Appelé quand une activité est rendue visible à l'utilisateur.,

Suivi par onResume() si l'activité passe à l'avant-plan ou onStop() si caché.

**11.** Quels énoncés sont vrais pour la méthode onStop() d'une activité?

La réponse correcte est : 

Appelé quand l'activité n'est plus visible pour l'usager.,

Sera suivi de onRestart() ou onDestroy() ou rien.,

Risque de ne jamais être appelé  si le système manque de ressources.

**12.** Quelle(s) méthode(s) permet(ent) de terminer une activité Android?

La réponse correcte est : finish(), finishActivity(int requestCode)

13. Quelle méthode d'activité doit-on utiliser pour récupérer la référence d'une vue Android en utilisant le "id" d'une ressource XML?

La réponse correcte est : findViewById(int id)

14. Comment passer des données entre d'une *Activity*à l'autre ?

La réponse correcte est : Intent

15. Que contient le manifeste d'application Android?

La réponse correcte est :

Des informations relatives à la mise-en page de l'application.,

Des informations relatives à l'icone de l'application.,

Tous les textes multilingues de l'application,

 Des informations relatives aux activités de l'application.,

Le nom du package principal des classes de l'application,,

Des informations relatives aux services de l'application.,

Des informations relatives aux permissions requises par l'application.,

Des informations relatives aux permissions requises par les autres applications qui veulent interagir avec l'application.

16. Qu’est-ce qu’un *Context* sur Android?

La réponse correcte est :

C'est utilisé pour stocker les objets communs d'une application.,

C'est utilisé pour instancier de nouvelles composantes comme les vues.,

C'est utilisé pour démarrer des activitées..,

C'est utilisé pour démarrer des services.

17. Le terme Activity dans le contexte de développement Android correspond à :

La réponse correcte est : Une classe Java

18. Que contient le fichier R.java?

La réponse correcte est : Le code généré par le SDK contenant tous les identifiants des ressources.

19. Que veut dire l'acronyme ADB?

La réponse correcte est : Android Debug Bridge

20. Quelle est la classe mère de tous les widgets sous Android?

La réponse correcte est : View

21. Quel est le nom du fichier à modifier pour demander la permission d'utiliser le GPS?

AndroidManifest.xml

22. Votre application Android s'appelle Quiz. Android studio vous a créé la structure de répertoire .../Quiz/app

Quelle est la suite du chemin pour trouver le code source de la classe ca.qc.cegepsth.Quiz.java? (utilisez / comme séparateur)

La réponse correcte est : /src/main/java/ca/qc/cegepsth/tapis

23. Votre application Android s'appelle Quiz. Android studio vous a créé la structure de répertoire .../Quiz/app

Vous utilisez des textes multilingues. Quelle est la suite du chemin pour trouver le fichier strings.xml ? (utilisez / comme séparateur)

/src/main/res/values/

24. Votre application Android s'appelle Quiz. Android studio vous a créé la structure de répertoire .../Quiz/app

Quelle est la suite du chemin pour trouver AndroidManifest.xml ? (utilisez / comme séparateur)

/src/main/

25. Les activités doivent être enregistrées dans le manifeste pour être exploitables.

La réponse correcte est « Vrai ».

26. Quand un appareil Android est connecté par ADB, je peux ouvrir une console qui supporte beaucoup de commandes standard en linux.

La réponse correcte est « Vrai ».

27. Les Intents me permettent d'ouvrir des activités d'une autre machine virtuelle Java.

En effet, puisque toutes les applications Android opèrent dans des machines virtuelles séparées, invoquer une activité d'une autre application permet de passer d'une machine virtuelle à l'autre.

La réponse correcte est « Vrai ».

**Quiz 3**

Les fragments doivent être enregistrés dans le manifeste pour être exploitable.

La réponse correcte est « Faux »

Associez les conventions de codes

La réponse correcte est :

Un nom de classe commence toujours par – une majuscule,

Un nom de "package" ne contient surtout pas – une majuscule,

Un nom d'argument commence toujours par – une minuscule,

Un nom de méthode commence toujours par – une minuscule,

Les espaces dans un nom sont remplacés par – une majuscule en milieu de mot,

Un nom d'attribut commence toujours par – une minuscule

Associez correctement le style approprié

La réponse correcte est :

Un attribut booléen est représenté idéalement par – un verbe d'état,

Une collection, liste, ... ou toute forme d'ensemble se caractérise par – le pluriel,

Une méthode est représentée idéalement par – un verbe d'action,

Une classe est représenté par – un nom commun

Si je fais un travail d'équipe, j'ai la même note que mes coéquipiers.

La réponse correcte est « Faux »