UNIVERSITÉ PARIS 7 DENIS DIDEROT - IF1 - PARTIEL - 29 OCTOBRE 2011 - 14^{00}-16^{00}

Aucun document. Aucune machine. Le barème est indicatif. Les cinq exercices sont indépendants.
Une question peut toujours être traitée en utilisant les précédentes (traitées ou non).
Les morceaux de code Java devront être clairement présentés, indentés et commentés.
Les classes `String`, `Deug`, `System` et `Scanner` sont interdites (sauf questions 2, 3 et 5 de l'exercice 4).

Exercice 1. (3 points) Expliquer ce que produit l'exécution de chacun des programmes suivants :

```
class Hello{
   public static void bonjour(String[] args){
      System.out.println("Hello World!");
   }
}
```

```
class MeKesDonk{
   public static void main(String[] args){
      int x, y=9, z=5;
      double s, t;
      x=y+2; System.out.println("x: "+x);
      y=y-2; System.out.println("x: "+x);
             System.out.println("y: "+y);
      z=y%z; System.out.println("z: "+z);
      y=y+9; System.out.println("y: "+y);
      x=4+z/y; System.out.println("x: "+x);
      s=4+z/y; System.out.println("s: "+s);
      s=y;
      t=4+z/s; System.out.println("t: "+t);
   }
}
```

```
class KrouA{
   static int f1(int n){
      if (n<=1 && n>=-1) return n;
      return f2(n/2);
   }

   static int f2(int n){
      if (n<=2 && n>=-2) return n;
      return f1(n+1);
   }

   public static void main(String[] args){
      System.out.println(f1(128));
   }
}
```

Exercice 2. (3 points) Expliquer chacun des messages d'erreur suivants :

```
MonControleur.java:15: ';' expected
        this.canvas.repaint()
                            ^
1 error
```

message 1

```
Test.java:7: essai(int) in Test cannot
             be applied to (double)
        essai(x);
        ^
1 error
```

message 2

```
Exception in thread "main" java.lang.NoSuchMethodError: main
```

message 3

```
Test.java:9: cannot return a value from method whose result type is void
        return n;
               ^
1 error
```

message 4

```
MaFenetre.java:38: fill(java.awt.Shape) in java.awt.Graphics2D cannot
                   be applied to (Canvas)
        ((Graphics2D) g).fill(this);
                        ^
1 error
```

message 5

```
Test.java:11: cannot find symbol
symbol : variable y
location: class Test
        System.out.print(y);
                         ^
1 error
```

message 6

```
Test.java:15: cannot find symbol
symbol : method sin(double)
location: class Test
        sin(x);
        ^
1 error
```

message 7

Une question peut toujours être traitée en utilisant les précédentes (traitées ou non).
Les classes `String`, `Deug`, `System` et `Scanner` sont interdites (sauf questions 2, 3 et 5 de l'exercice 4).

Exercice 3. (4 points) Au musée de Trevors, les tarifs d'entrée sont définis par les règles suivantes :
- pour les billets individuels : gratuité pour les enfants d'au plus 3 ans, tarif réduit (3€) pour les enfants entre 4 et 10 ans et les personnes âgées à partir de 65 ans et tarif plein (5€) pour les autres.
- pour les billets de groupes (au moins 8 personnes) : sans considération d'âge, 10% de réduction sur le tarif plein pour les groupes jusqu'à 15 personnes incluses, 20% au-delà.

Écrire des fonctions `tarifIndividu` et `tarifGroupe` renvoyant le prix à payer à partir d'arguments pertinents.

Exercice 4. (6 points) On considère un jeu de dé pour deux joueurs (identifiés par `1` et `2`) dont voici le principe :
- une partie est composée de 3 tours
- à chaque tour, chacun des deux joueurs peut choisir de jeter ou non un dé à 6 faces
- en fin de partie, le score d'un joueur est la somme des valeurs obtenues lors de ses jets de dé
- un joueur gagne si son score est inférieur ou égal à 14 et strictement supérieur au score de l'autre joueur
- si les scores sont égaux ou sont tous les deux strictement supérieurs à 14, aucun des joueurs ne gagne

1. Écrire une fonction `jetDeDe` qui renvoie un entier tiré au hasard entre `1` et `6`. On pourra utiliser la fonction `random` de la classe `Math` qui renvoie un réel aléatoire supérieur ou égal à `0` et strictement inférieur à `1` et
   - soit utiliser la fonction `partieEntiere` de la classe `Chapitre1` définie en cours-td,
   - soit découper l'intervalle `[0,1[` en six intervalles de même longueur.
2. Écrire une fonction `tourJoueur` qui prend un numéro `k` de joueur en argument, demande à ce joueur s'il souhaite jeter le dé, auquel cas renvoie la valeur d'un jet de dé et renvoie `0` sinon.
3. Écrire une fonction `annoncerValeurs` qui prend en arguments deux valeurs `v1` (valeur du jet du joueur `1`) et `v2` (valeur du jet du joueur `2`) et affiche un message annonçant la valeur associée à chaque joueur.
4. Écrire une fonction `gagnant` qui prend en arguments deux scores `s1` (score du joueur `1`) et `s2` (score du joueur `2`) et renvoie le numéro du joueur gagnant s'il existe ou `0` sinon.
5. Écrire une fonction `partie` mettant en œuvre le jeu et utilisant les quatre fonctions précédentes. Cette fonction demande à chaque tour à chacun des joueurs (identifiés par `1` et `2`) s'il souhaite lancer le dé (réponse `1`) ou pas (réponse `0`) ; une fois que chacun des joueurs a choisi, la fonction affiche la valeur du jet de dé pour chacun des joueurs pour ce tour et met à jour les scores. À la fin du troisième tour, la fonction affiche un message annonçant le gagnant s'il existe, ou un message indiquant qu'aucun des joueurs n'a gagné sinon.

Exercice 5. (6 points) Soit `n` un entier compris entre `2` et `9`. Un entier positif `p` est appelé `n`-*parasite* si le produit `p`$\times$`n` est égal à la rotation de `p`, c'est-à-dire l'entier dont l'écriture est obtenue à partir de celle de `p` en déplaçant le chiffre des unités au début. Par exemple, `142857` est `5`-parasite car `142857`$\times$`5` vaut `714285`.

1. Écrire une fonction récursive `puissance` qui prend en arguments deux entiers positifs `a` et `b` et renvoie l'entier `a` élevé à la puissance `b`.
2. On définit la fonction récursive `ordre` comme suit :

```
static int ordre(int n, int m){
    if (puissance(10,m)%(10*n-1)==1) return m;
    return ordre(n,m+1);
}
```

   Expliquer ce que produit l'évaluation de `ordre(4,2)`.
3. Écrire une fonction `unParasite` qui prend en argument en entier `n` compris entre `2` et `9` et renvoie l'entier $\frac{10^m-1}{10n-1}n$ où `m` est l'entier renvoyé par la fonction `ordre` appelée avec les paramètres `n` et `2`. On sait que cet entier est alors `n`-parasite.
4. Écrire une fonction récursive `nbChiffres` qui prend en argument un entier positif `a` et renvoie le nombre de chiffre(s) dans l'écriture de `a`.
5. Écrire une fonction `rotation` qui prend en argument un entier positif `p` et renvoie l'entier dont l'écriture est obtenue à partir de celle de `p` en déplaçant le chiffre des unités au début.
6. Écrire une fonction `estParasite` qui prend en argument un entier `n` compris entre `2` et `9` et un entier positif `p` et teste si `p` est `n`-parasite.