

1. Naissance d'un programme

Exercice 1 : Pour se familiariser avec Linux

Le but de cet exercice est de se familiariser avec linux afin que ce ne soit pas un frein à la programmation. Le détail de toutes les commandes s'obtient en tapant :

```
$ man <NomCommande>
```

Par ailleurs, un poly sur Unix figure à l'adresse suivante :

<http://irfu.cea.fr/Phoce/Cours/index.php>

1. A l'aide de la commande *mkdir* créer un répertoire JavaTest
2. Dans ce répertoire, lancer un éditeur de texte comme *nedit* et créer un fichier texte *test.txt*.
Modifier ce fichier.
On remarquera que le caractère *&* permet de lancer un processus en tache de fond
3. A l'aide de la commande *ls* lister les fichiers existants dans votre répertoire JavaTest.
On remarquera que le caractère ***, respectivement *?* indique que toute chaine de caractères peut se substituer à l'***, respectivement que tout caractères peut se substituer à *?*.
4. Copier ce fichier avec la commande *cp*
5. Effacer le précédant fichier avec la commande *rm*
6. Créer de nouveau répertoire et naviguer dans les répertoires avec la commande *cd*

Exercice 2 : compiler

Télécharger les fichiers *bonjour.c*, *bonjour.cpp*, *bonjour.class* qui se trouvent à l'adresse suivante :

<http://irfu.cea.fr/Phoce/Cours/index.php>

Compiler et exécuter ces fichiers

2. Java, les bases

Exercice 3 : un exemple simple

En utilisant *éclipse*, créer un projet *exemple* qui permet d'écrire, compiler et exécuter le programme suivant :

```
public class HelloWorld
{
    public static void main(String [] args)
    {
        System.out.println("*****");
        System.out.println("*  bonjour  *");
    }
}
```

```
System.out.println("*****");
}
}
```

Exercice 4 : comprendre les mécanismes d'affectation, d'échange de valeur, de cast

1. Ecrire un programme qui affiche les valeurs de a, b, c après l'exécution de l'extrait de programme suivant :

```
float a = 3.5 ;
float b = 1.5 ;
float c ;

c = a + b ;
b = a + c ;
a = b ;
```

Interpréter les résultats obtenus

2. Même question avec l'extrait de programme suivant :

```
int a = 5 ;
int b = 7 ;

a = b ;
b = a ;
a = a + b ;
b = a - b ;
a = a - b ;
```

Interpréter les résultats obtenus

3. Même question avec l'extrait de programme suivant :

```
int valeur = 7, chiffre = 2, i1, i2 ;
float f1, f2 ;
i1 = valeur/chiffre ;
i2 = chiffre/valeur ;
f1 = (float) (valeur/chiffre) ;
f2 = (float) (valeur/chiffre) + 0.5f ;
i1 = (int) f1 ;
i2 = (int) f2 ;
f1 = (float) valeur / (float) chiffre ;
f2 = (float) valeur / (float) chiffre + 0.5f ;
i1 = (int) f1 ;
i2 = (int) f2 ;
```

Interpréter les résultats obtenus

Exercice 5

Ecrire un programme qui lit successivement 2 entiers et renvoie leur somme. Modifier le programme pour 2 flottants, puis pour un entier et un flottant (deux solutions)

Exercice 6 : calculatrice franc/euros

En prenant exemple sur la structure suivante, écrire un programme Euro qui convertit des francs en euros. On rappelle que 1 euro = 6.55957 francs.

```
public class.....//donner le nom de la classe
{
    public static void main(String [] argument)
    {
        //Declarer les variables representants les francs et les euros
        // ainsi que le taux de conversion
        .....

        // Afficher et saisir le nombre de francs
        .....

        //Calculer le nombre d'euros
        .....

        //Afficher le resultat
    }
}
```

Exercice 7

1. Déclarer un tableau de 10 entiers
2. Le remplir avec les entiers pairs
3. Afficher à l'écran les valeurs du tableau

3. Structure de programmation

Exercice 8 : Le temps qu'il fait...

Soit la température T. Ecrire une classe qui affiche :

- froid si $T < 8$
- frais si $8 < T < 17$
- bon si $17 < T < 25$
- chaud sinon

Exercice 9 : Calculatrice

En utilisant la structure switch, écrire un programme qui simule une machine à calculer dont les opérations sont l'addition (+), la soustraction (-), la multiplication (*) et la division (/).

1. Le programme demande à l'utilisateur d'entrer deux valeurs puis le caractère correspondant à l'opération à effectuer. Le programme affiche l'opération effectuée ainsi que le résultat
2. Saisissez dans cet ordre 2, 0 puis /. Que se passe-t-il ? Pourquoi ?
3. Modifier le programme de façon à ne plus rencontrer cette situation.

Exercice 10 : Compteur

Réaliser une fonction qui affiche à l'écran les entiers de 0 à 100 compris en utilisant trois types de boucle (while,do while,for).

Exercice 11 : Devinette

Ecrire un programme devinette qui tire un nombre entier au hasard entre 0 et 10 et demande à l'utilisateur de trouver ce nombre. La méthode est la suivante :

1. Tirer un nombre au hasard entre 0 et 10. Pour tirer un nombre entier au hasard, on pourra utiliser :

```
i = (int) ( 10*Math.random() );
```

où i est une variable entière qui reçoit la valeur tirée au hasard

2. Lire un nombre
3. Tant que le nombre lu est différent du nombre tiré au hasard :
 - lire un nombre
 - compter le nombre de boucle
4. Afficher un message de réussite ainsi que le nombre de boucles

4. Classes, méthodes et objets

Exercice 12 : utilisation de fonction

```
public class Fonction
{
    public static void main(String [] parametre)
    {
```

```

//Declaration des variables
int a, compteur;

for (compteur = 0; compteur <= 5; compteur ++)
{
    a = calculer(compteur);
    System.out.print(a + " a ");
}
} //fin du main

public static int calculer(int x)
{
    int y;

    y = x*x;
    return y;
} //fin de calculer
} // fin de classe

```

1. Délimiter les 3 blocs délimitant la fonction main(), la fonction calculer() et la classe Fonction
2. Quel est le paramètre formel de la fonction calculer() ?
3. Quelles sont les valeurs effectives de la fonction calculer() lors de son appel depuis la fonction main ?
4. Quels sont les résultats produits par la fonction calculer() ?
5. Quelles sont les valeurs transmises à la variable a ?
6. Décrire l'affichage réalisé par la fonction main().

Exercice 13

```

public static int f(int x)
{
    int resultat ;

    resultat = -x*x+3*x-2 ;
    return resultat;
}

```

1. Ecrire la fonction main() qui affiche le résultat de la fonction f(x) pour x = 0.
2. Transformer la fonction main() de façon à calculer et à afficher le résultat de la fonction pour x variant entre -5 et 5.
3. Déterminer le maximum en le stockant dans une variable max

Exercice 14

1. Ecrire une fonction `verifier()` qui demande à l'utilisateur la saisie d'une valeur tant que cette valeur est comprise entre 0 et 100. Quand la valeur appartient à l'intervalle demandé, la fonction retourne la valeur saisie.
2. Comment modifier la fonction `verifier` pour que la valeur saisie ne soit plus entre 0 et 100 mais entre deux paramètres `a` et `b`.

Exercice 15 : des triangles...

```
import java.util.*;
public class Triangle {
    public int xA, yA, xB, yB, xC, yC ;

    public void créer()
    {
        Scanner lectureClavier = new Scanner(System.in);
        System.out.println("Point A : ");
        System.out.print("Entrez l'abscisse : ");
        xA = lectureClavier.nextInt();
        System.out.print("Entrez l'ordonnee : ");
        yA = lectureClavier.nextInt();
        System.out.println("Point B : ");
        System.out.print("Entrez l'abscisse : ");
        xB = lectureClavier.nextInt();
        System.out.print("Entrez l'ordonnee : ");
        yB = lectureClavier.nextInt();
        System.out.println("Point C : ");
        System.out.print("Entrez l'abscisse : ");
        xC = lectureClavier.nextInt();
        System.out.print("Entrez l'ordonnee : ");
        yC = lectureClavier.nextInt();
    }

    public void afficher() {
        System.out.println("Point A : " + xA + " " + yA);
        System.out.println("Point B : " + xB + " " + yB);
        System.out.println("Point C : " + xC + " " + yC);
    }

    public void deplacer(int nx, int ny){
        xA += nx;
        yA += ny;
        xB += nx;
        yB += ny;
        xC += nx;
        yC += ny;
    }
} // Fin de la classe Triangle

public class FaireDesTriangles {
```

```

public static void main(String[] args) {
    Triangle T = new Triangle();
    T.afficher();
    T.créer();
    T.afficher();
    T.deplacer(100, 0);
    T.afficher();
}
}

```

1. Quel est le programme qui correspond à l'application ?
2. Quel est le programme définissant le type Triangle
3. Rechercher les attributs de la classe Triangle et donner leur nom.
4. Combien de méthodes sont définies dans la classe Triangle ? Donner leur nom.
5. Quels sont les objets utilisés par l'application FaireDesTriangles ? Que valent leurs données xA, yA, xB, yB, xC et yC après exécution de l'instruction déclaration ?

Exercice 16 : créer une bibliothèque

L'objectif est de définir une représentation d'un objet livre.

1. Sachant qu'un livre est défini à partir de son titre, du nom et du prénom de l'auteur, d'une catégorie (Policier, Roman, Junior, Science-fiction...), d'un numéro d'ISBN et d'un code alphanumérique unique (voir question 5), définir les données de la classe Livre
2. Ecrire une application bibliothèque qui utilise un objet livrePoche de type Livre et qui demande la saisie au clavier du titre, des nom et prénom de l'auteur et du numéro ISBN
3. Dans la classe Livre, ajouter une méthode afficherUnLivre() qui affiche les caractéristiques du livre concerné
4. Modifier l'application Bibliotheque de façon à afficher les caractéristiques de l'objet livrePoche
5. Le code d'enregistrement d'un livre est construit à partir de deux premières lettres des nom et prénom de l'auteur, de la catégorie du livre et des deux derniers chiffres du code ISBN. Ecrire la méthode calculerCode() qui permet de calculer ce code.
6. Modifier l'application Bibliotheque de façon à calculer et afficher le code de l'objet livrePoche