**Classes et techniques utilitaires**

**La classe système**

Est une classe qui offres des propriete et des methodes, cette classe ne peut pas etre instancier car il n’a pas de constructeur visible.

Système. Out.printLn

Système. err.printLn

Système. in.printLn ; in : pour exploiter ce qui est entrer sur clavier …

Système.exit() ; pour mettre fin au programme

long time=Système.currentTimeMilles() ; pour données combien de temps est ecouler depuis 1970 en millisecondes

système.nanoTime permet de donner le temp en nanoseconde permet de donner les ecart du temp .

**long timeStat=** système.nanoTime() ;

system.out.printLn= (‘hello world’) ;

**long timeFin=** système.nanoTime() ;

system.out.printLn= (eimeStart -timeFin) ;

ce code donne en résulta combien de temp à pris l’affichage de ‘Hello world’ ; en nanotime

system.out.printLn(system.GetProperties()) ;

permet d’afficher les propriétés du système.

**La classe scanner**

Permet de scanner ce qui vas etre entre dans le clavier ou …

Exemple d’utilisation

import java.util.Scanner; pour pouvoir utiliser la classe Scanner  
  
public class Helloworld {  
  
 public static void main(String[] args){  
  
 System.*out*.println("Hello World");  
 Scanner sc=new Scanner(System.*in*);  
 String ligneSivant=sc.nextLine(); après l’affichage de hello world le système va nous donner la main pour ecrire au clavier   
 System.*out*.println("la ligne suivante est : "+ligneSivant);l’affichage de ce qu’on a taper par le clavier  
 }  
}

exemple 2 exercice video 88

import java.util.Scanner;  
  
public class HelloUniverse {  
  
 public static void main(String... args) {  
  
  
 Vaisseau fregat = new VaisseauDeGuerre("fregat");  
  
 Vaisseau croiseur = new VaisseauDeGuerre("CROISEUR");  
 Vaisseau chasseur = new VaisseauDeGuerre("CHASSEUR");  
 Vaisseau cargo = new VaisseauDeGuerre("CARGO");  
 Vaisseau veasseauMonde = new VaisseauDeGuerre("Terre");  
  
 Scanner sc=new Scanner(System.*in*);  
 System.*out*.println("quel vaisseau choisissez vous ? ");  
 String vaisseauChoisi=sc.nextLine();  
 System.*out*.println("sur quel planete ? ");  
 String planeteChoisi=sc.nextLine();  
 System.*out*.println("quel tonnage ? ");  
 int tonnageChoisi=sc.nextInt();  
  
 Vaisseau vaisseau=null;  
 switch (vaisseauChoisi){  
 case "fregat":  
 vaisseau=fregat;  
 break;  
 case "chasseur":  
 vaisseau=chasseur;  
 break;  
 case "cargo":  
 vaisseau=cargo;  
 break;  
 }  
 PlaneteTellurique planeteC=null;  
 switch (planeteChoisi){  
 case "mars":  
 vaisseau=cargo;  
 break;  
 case "uranus":  
 vaisseau=cargo;  
 break;  
 }

**Classes conteneur**

Encapsule chaque type premitif par exemple pour int 🡪Integer ; string🡪String …

Chaque classe conteneur possede une methode permettant de definir la valeur similair d’un variable de string le rendre int par exemple.

L’un des avantage ces de beneficier de tous les caracterestique des classe en génerale par contre au variable de type primitif par exemple les classe onteneur peuvent etre nul

Exemple d’utilisation

Integer i = new Integer(12);  
  
Float f=new Float(12.2f);  
Float f2=new Float("62.8");  
float f3=f.floatValue();  
int i2=i.intValue();  
float f4=Float.*parseFloat*("89.9");

L’auto boxing et auto-unboxing : permet de donner une valeur a une variable de classe conteneur sans instancier le variable déelarer . c’est de changer un variable de type conteneur a un variable de type primitif

Integer a = 5 ; a la place de Integer a = new Integer(5) ;

**Conversions de type(parseInt, parseString …)**

Pour transformer une chaine en int 🡪 int i3=Integer.parseInt(’16 ‘) ;

Exp

int i3=Integer.*parseInt*(5); transferer Integer en Int  
String s4=String.*valueOf*(i3); le meme caspour transferer un long

Long long1 = *i3.ongvalueOf*(i3)

**Les types enumerer**

Il s’agit de restrin les valeurs d’une variable

Par exemple si nous avons une boite a vitesse qui contient troi type :

Auto , manuel, semi-auto.

A la place de déclarer un boolean il faut un autre type ou de crées une classe exp

public class TypeBoitVitesse {  
 static final int *auto*=1; le mot clé final indique que cette variable est fixe  
 static final int *manuel*=2;  
 static final int semi-*auto*=3;  
}

Voiture peugeot=new Voiture();  
peugeot.typB=TypeBoitVitesse.*auto*; ou peugeot.typB=1;  
if (peugeot.typB==TypeBoitVitesse.*auto*){  
 System.*out*.println("la boite a vitesse est auto");  
}

Resulta

la boite a vitesse est auto

dans ce cas on va utiliser le système enumeration

a la place de declarer une classe avec des variable static final on changer le type de classe en enum

public enum TypeBoitVitesse {  
 *auto*,  
 *manuel*,  
 *semiAuto*;  
}

public class Voiture extends Vehicule implements Videngeable {  
  
 TypeBoitVitesse typB;

public class origin {  
 public static void main(String[] args){  
  
 Voiture peugeot=new Voiture();  
 peugeot.typB=TypeBoitVitesse.*auto*;  
 switch (peugeot.typB){  
 case *auto*:  
 System.*out*.println("la boite est automatique");  
 break;  
 case *manuel*:  
 System.*out*.println("la boite est manuel");  
 break;  
 case *semiAuto*:  
 System.*out*.println("la boite est semi\_auto");  
 break;  
 }

résulta

la boite est automatique

on peut ajouter un indication apres la declaration des enums exp

public enum TypeBoitVitesse {  
 *auto*("automatique"),  
 *manuel*("semi-manuelle"),  
 *semiAuto*("Manuelle");  
 String nomTypeBoite;  
 // java essaie de trouver un constructeur avec strin en argument: on va l'jouter  
 private TypeBoitVitesse(String nomTypeBoite){  
 this.nomTypeBoite=nomTypeBoite;  
 }  
}

public class Voiture extends Vehicule implements Videngeable {  
  
 TypeBoitVitesse typB;

public class origin {  
 public static void main(String[] args){  
  
 Voiture peugeot=new Voiture();  
 peugeot.typB=TypeBoitVitesse.*auto*;  
 System.*out*.println("le type de ma boite vitesse est : "+peugeot.typB.nomTypeBoite);

System.*out*.println("le type de ma boite vitesse est : "+peugeot.typB.ordinal());cette ligne permet de donner le numero d’ordre du variable typeB

}  
}

résulta

la boite est automatique

NB : le meme résulta que le code précédent

Exercice Final a refaire

QUIZ

Il y a bien une méthode pour chaque type : nextLine() pour les String, nextInt() pour les int etc... pour les char; il faudra faire un nextLine().chartAt(0)

Sytem.out C'est en effet le canal de sortie par défaut qui se trouve être la console en général mais ceci est modifiable.