

Correction TP2 – Types Génériques

Partie 1 :

```
public class Pile<Element> {  
  
    private Element T[]= (Element[])new Object [100];  
    private int sommet;  
    public Pile()  
    {sommet=-1;}  
    public void empiler(Element elt){sommet++;T[sommet]=elt;}  
    public void depiler(){sommet--;}  
    public Element som(){return T[sommet];}  
    public void vider(){sommet=-1;}  
    public void afficher(){  
        for (int i=0;i<=sommet; i++)  
            System.out.println(T[i]+" ");}  
    public boolean pileVide()  
    {return (sommet==-1);}  
    public boolean pilePleine()  
    {return (sommet== T.length-1);}  
}
```

```
import java.util.Scanner;  
public class TestPile {  
  
    public static void main(String[] args) {  
        Pile<String> p=new Pile<String>();  
        int choix;  
        Scanner sc=new Scanner(System.in);  
        do{  
            System.out.println("1-empiler\n2-depiler\n3-lire la valeur au sommet\n4-afficher\n5-  
vider\n6-quitter\nDonnez votre choix");  
            choix =sc.nextInt();  
            switch(choix)  
            {  
                case 1: System.out.println("donnez l'élément à empiler");  
                    String elt=sc.next();  
                    if (!p.pilePleine())p.empiler(elt);  
                    else System.out.println("pile pleine");  
                    break;  
                case 2:if (!p.pileVide())p.depiler();  
                    else System.out.println("pile vide");  
                    break;  
                case 3:if (!p.pileVide())System.out.println("la valeur au sommet est:"+p.som());  
                    else System.out.println("pile vide");  
                    break;  
                case 4:if (!p.pileVide())p.afficher();  
            }  
        }  
    }  
}
```

```

        else System.out.println("pile vide");
        break;
        case 5: if (!p.pileVide()) p.vider();
        else System.out.println("pile vide");
        break;
    } } while(choix!=6);

}

}

```

Partie2 :

```

import java.util.ArrayList;
import java.util.Iterator;

public class Pile {
    private ArrayList<Integer> liste = new ArrayList<Integer>();
    public void empiler(int n) {liste.add(n);}
    public void depiler() {liste.remove(liste.size() - 1);}
    public boolean pileVide() {return liste.size() == 0;}
    public int som() {return liste.get(liste.size()-1);}
    public void vider() {liste.clear();}
    public void afficher(){
        /*for (int i=0;i<liste.size(); i++)
            System.out.println(liste.get(i)+" ");*/

        Iterator <Integer> it=liste.iterator();
        while (it.hasNext())
            System.out.println(it.next()+" ");
    }
}

```

```

import java.util.Scanner;
public class TestPile {

    public static void main(String[] args) {
        Pile p=new Pile();
        int choix;
        Scanner sc=new Scanner(System.in);
        do{

```

```

        System.out.println("1-empiler\n2-depiler\n3-lire la valeur au sommet\n4-afficher\n5-
vider\n6-quitter\nDonnez votre choix");
        choix =sc.nextInt();
        switch(choix)
        {case 1: System.out.println("donnez l'élément à empiler");
            int elt=sc.nextInt();
            p.empiler(elt);

            break;
        case 2: if (!p.pileVide())p.depiler();
        else System.out.println("pile vide");
            break;
        case 3: if (!p.pileVide())System.out.println("la valeur au sommet est:"+p.som());
        else System.out.println("pile vide");
            break;
        case 4: if (!p.pileVide())p.afficher();
        else System.out.println("pile vide");
            break;
        case 5: if (!p.pileVide())p.vider();
        else System.out.println("pile vide");
            break;
        } } while(choix!=6);
    }
}

```

Partie3 :

```

import java.util.ArrayList;

public class Pile <Element>{
    private ArrayList <Element>liste = new ArrayList<Element>() ;
    public void empiler(Element n) {liste.add(n);}
    public void depiler() {liste.remove(liste.size() - 1);}
    public boolean pileVide() {return liste.size() == 0;}
    public Element som(){return liste.get(liste.size()-1);}
    public void vider(){liste.clear();}
    public void afficher(){
        for (int i=0;i<liste.size(); i++)
            System.out.println(liste.get(i)+" ");}
}

import java.util.Scanner;
public class TestPile {

    public static void main(String[] args) {
        Pile<Integer> p=new Pile<Integer>();
        int choix;
    }
}

```

```

Scanner sc=new Scanner(System.in);
do{
    System.out.println("1-empiler\n2-depiler\n3-lire la valeur au sommet\n4-afficher\n5-
vider\n6-quitter\nDonnez votre choix");
    choix =sc.nextInt();
    switch(choix)
    {case 1: System.out.println("donnez l'élément à empiler");
        int elt=sc.nextInt();
        p.empiler(elt);

        break;
    case 2:if (!p.pileVide())p.depiler();
    else System.out.println("pile vide");
        break;
    case 3:if (!p.pileVide())System.out.println("la valeur au sommet est:"+p.som());
    else System.out.println("pile vide");
        break;
    case 4:if (!p.pileVide())p.afficher();
    else System.out.println("pile vide");
        break;
    case 5:if (!p.pileVide())p.vider();
    else System.out.println("pile vide");
        break;
    } }while(choix!=6);
}
}

```