

Tan
Kenthvuth

TP java héritage

1) Concevez la hiérarchie de classes suivantes :

> Membres donnés à prévoir pour chaque classe :

- **Ordinateur** : référence, processeur, RAM, disque dur, système d'exploitation et prix de vente,
- **Netbook** : taille écran, poids, webcam intégrée (O/N), lecteur de carte mémoire (O/N)
- **Portable** : taille écran, poids, webcam intégrée (O/N), lecteur de DVD (O/N)
- **OrdiBureau** : carte graphique, chipset, clavier

> Chaque classe doit posséder un constructeur obligeant l'utilisateur à renseigner toutes les propriétés d'un objet.

> Chaque classe doit posséder une fonction membre permettant l'affichage de l'ensemble des propriétés d'un objet (nommez-là toString).

Classe Ordinateur (classe mère):

```
1 package tan_tp_java_heritage;
2
3 import java.util.ArrayList;
4
5 public class Ordinateur {
6
7     //Attributs
8     private String reference;
9     private String processeur;
10    private int ram;
11    private String disqueDur;
12    private String sysExploit;
13    private double prix;
14    protected ArrayList<Ordinateur> collection = new ArrayList<Ordinateur>();
15
16    public Ordinateur(String ref, String proc, int ram, String disque, String sys, float prix) {
17        this.reference = ref;
18        this.processeur = proc;
19        this.ram = ram;
20        this.disqueDur = disque;
21        this.sysExploit = sys;
22        this.prix = prix;
23    }
24
25    public String toString() {
26        return this.reference + " - " + this.processeur + " - " + this.ram + "Go RAM - " + this.disqueDur + " - " + this.sysExploit + " - " + this.prix;
27    }
28
29 }
30
```

Classe Netbook:

```
1 package tan_tp_java_heritage;
2
3
4 public class Netbook extends Ordinateur {
5
6     //Attributs
7     private float taille; // Taille en pouce
8     private float poids; // En kilogrammes
9     private boolean webcam;
10    private boolean lecteurCarteMemoire;
11
12    public Netbook(String ref, String proc, int ram, String disque, String sys, float prix, float taille, float poids, boolean webcam, boo
13        super(ref, proc, ram, disque, sys, prix);
14        this.taille = taille;
15        this.poids = poids;
16        this.webcam = webcam;
17        this.lecteurCarteMemoire = carteMemoire;
18    }
19
20    public String toString() {
21        String infos = super.toString();
22        infos = infos + " - " + this.taille + "'' - " + this.poids + "kg - " + this.webcam + " - " + this.lecteurCarteMemoire;
23        return infos;
24    }
25 }
26
27
```

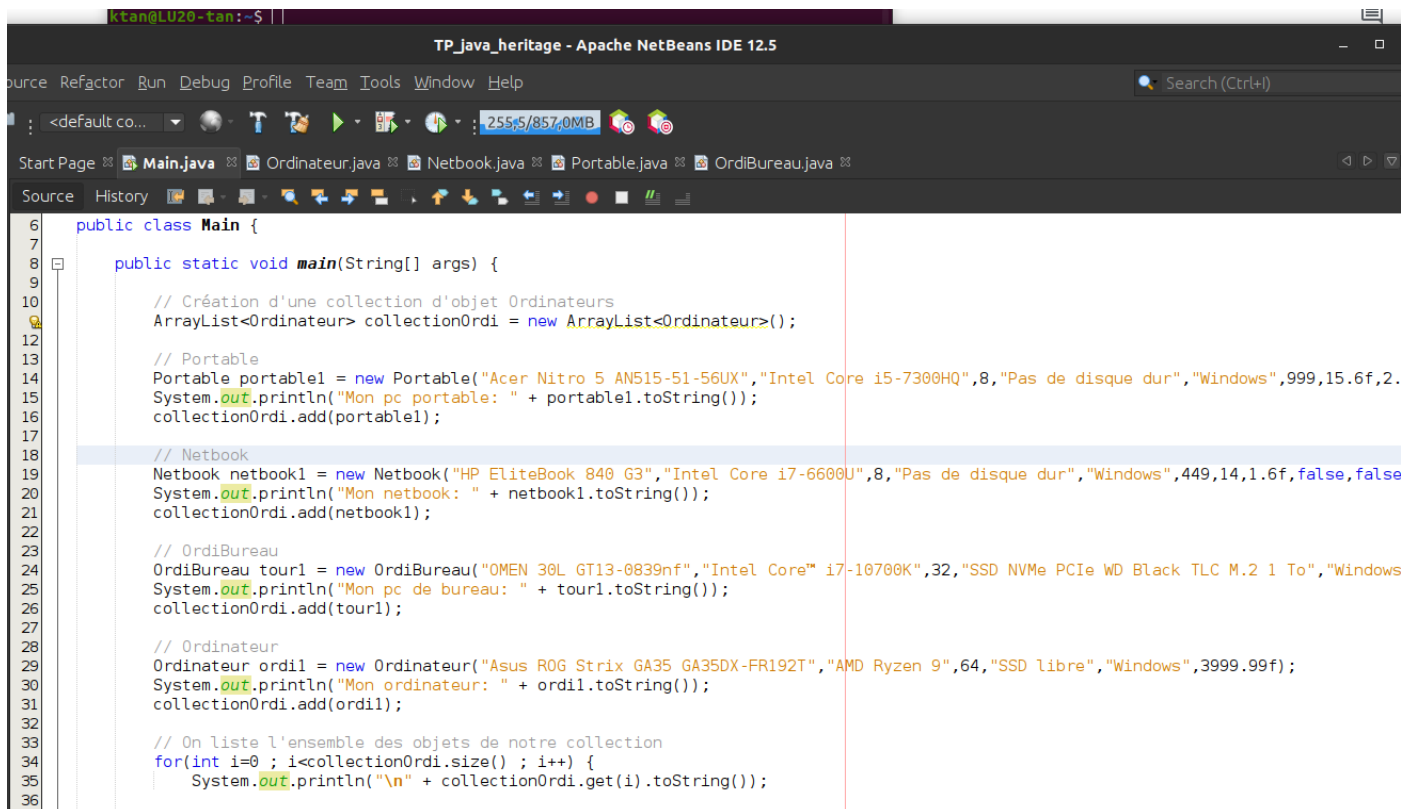
Classe Portable:

```
1 package tan_tp_java_heritage;
2
3
4 public class Portable extends Ordinateur {
5
6     //Attributs
7     private float taille; // Taille en pouce
8     private float poids; // En kilogrammes
9     private boolean webcam;
10    private boolean lecteurDVD;
11
12    public Portable(String ref, String proc, int ram, String disque, String sys, float prix, float taille, float poids, boolean webcam, bo
13        super(ref, proc, ram, disque, sys, prix);
14        this.taille = taille;
15        this.poids = poids;
16        this.webcam = webcam;
17        this.lecteurDVD = lecteur;
18    }
19
20    public String toString() {
21        String infos = super.toString();
22        infos = infos + " - " + this.taille + "'' - " + this.poids + "kg - " + this.webcam + " - " + this.lecteurDVD;
23        return infos;
24    }
25 }
26
27
28
```

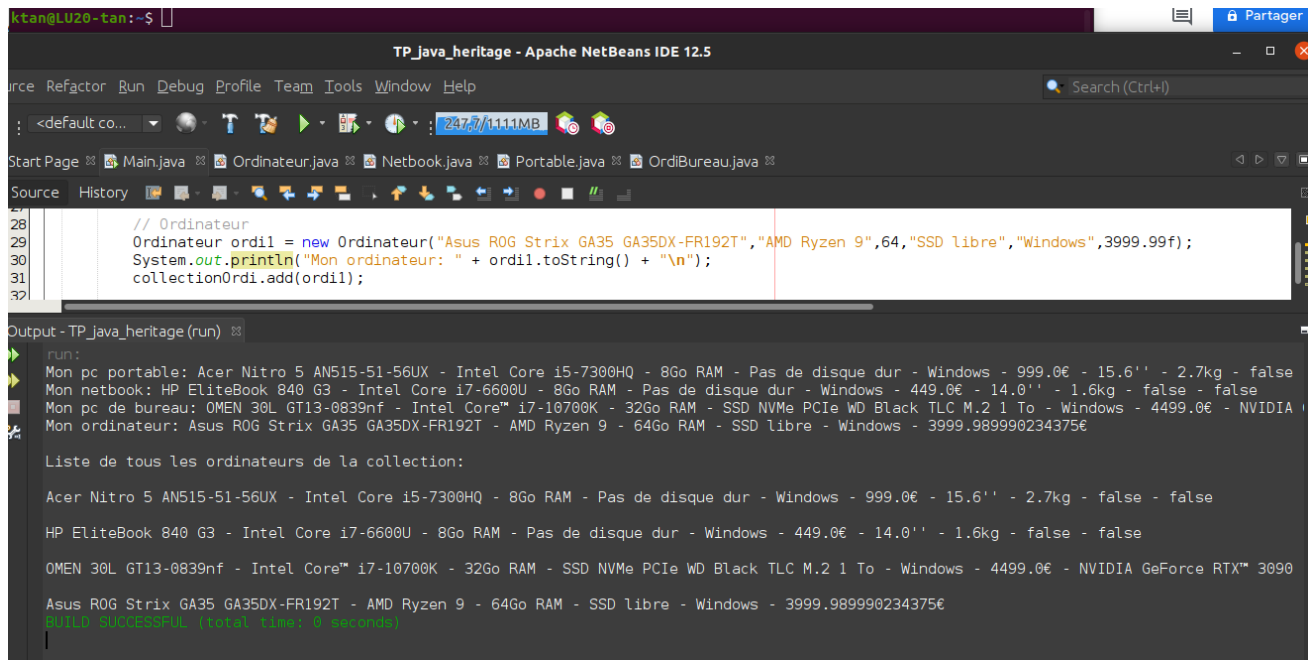
Classe OrdiBureau:

```
1 package tan_tp_java_heritage;
2
3
4 public class OrdiBureau extends Ordinateur {
5
6     //Attributs
7     private String carteGraphique;
8     private String chipset;
9     private String clavier;
10
11     public OrdiBureau(String ref, String proc, int ram, String disque, String sys, float prix, String cg, String chipset, String clavier) {
12         super(ref, proc, ram, disque, sys, prix);
13         this.carteGraphique = cg;
14         this.chipset = chipset;
15         this.clavier = clavier;
16     }
17
18     public String toString() {
19         String infos = super.toString();
20         infos = infos + " - " + this.carteGraphique + " - " + this.chipset + " - " + this.clavier;
21         return infos;
22     }
23
24 }
25
```

2) Testez votre réalisation en utilisant une collection. Votre programme devra tester l'ensemble des fonctionnalités pour les différentes classes, remplir la collection d'objets de différents types et afficher son contenu en fin de saisie.



```
6 public class Main {
7
8     public static void main(String[] args) {
9
10         // Création d'une collection d'objet Ordinateurs
11         ArrayList<Ordinateur> collectionOrdi = new ArrayList<Ordinateur>();
12
13         // Portable
14         Portable portable1 = new Portable("Acer Nitro 5 AN515-51-56UX", "Intel Core i5-7300HQ", 8, "Pas de disque dur", "Windows", 999, 15.6f, 2);
15         System.out.println("Mon pc portable: " + portable1.toString());
16         collectionOrdi.add(portable1);
17
18         // Netbook
19         Netbook netbook1 = new Netbook("HP EliteBook 840 G3", "Intel Core i7-6600U", 8, "Pas de disque dur", "Windows", 449, 14, 1.6f, false, false);
20         System.out.println("Mon netbook: " + netbook1.toString());
21         collectionOrdi.add(netbook1);
22
23         // OrdiBureau
24         OrdiBureau tour1 = new OrdiBureau("OMEN 30L GT13-0839nf", "Intel Core™ i7-10700K", 32, "SSD NVMe PCIe WD Black TLC M.2 1 To", "Windows");
25         System.out.println("Mon pc de bureau: " + tour1.toString());
26         collectionOrdi.add(tour1);
27
28         // Ordinateur
29         Ordinateur ordi1 = new Ordinateur("Asus ROG Strix GA35 GA35DX-FR192T", "AMD Ryzen 9", 64, "SSD libre", "Windows", 3999.99f);
30         System.out.println("Mon ordinateur: " + ordi1.toString());
31         collectionOrdi.add(ordi1);
32
33         // On liste l'ensemble des objets de notre collection
34         for(int i=0; i<collectionOrdi.size(); i++) {
35             System.out.println("\n" + collectionOrdi.get(i).toString());
36         }
37     }
38 }
```



```
TP_Java_heritage - Apache NetBeans IDE 12.5
Source Refactor Run Debug Profile Team Tools Window Help
: <default co... 247.7/1111MB
Start Page Main.java Ordinateur.java Netbook.java Portable.java OrdiBureau.java
Source History
28 // Ordinateur
29 Ordinateur ordi1 = new Ordinateur("Asus ROG Strix GA35 GA35DX-FR192T", "AMD Ryzen 9", 64, "SSD libre", "Windows", 3999.99f);
30 System.out.println("Mon ordinateur: " + ordi1.toString() + "\n");
31 collectionOrdi.add(ordi1);
32
Output - TP_Java_heritage (run)
run:
Mon pc portable: Acer Nitro 5 AN515-51-56UX - Intel Core i5-7300HQ - 8Go RAM - Pas de disque dur - Windows - 999.0€ - 15.6'' - 2.7kg - false
Mon netbook: HP EliteBook 840 G3 - Intel Core i7-6600U - 8Go RAM - Pas de disque dur - Windows - 449.0€ - 14.0'' - 1.6kg - false - false
Mon pc de bureau: OMEN 30L GT13-0839nf - Intel Core™ i7-10700K - 32Go RAM - SSD NVMe PCIe WD Black TLC M.2 1 To - Windows - 4499.0€ - NVIDIA
Mon ordinateur: Asus ROG Strix GA35 GA35DX-FR192T - AMD Ryzen 9 - 64Go RAM - SSD libre - Windows - 3999.989990234375€

Liste de tous les ordinateurs de la collection:

Acer Nitro 5 AN515-51-56UX - Intel Core i5-7300HQ - 8Go RAM - Pas de disque dur - Windows - 999.0€ - 15.6'' - 2.7kg - false - false
HP EliteBook 840 G3 - Intel Core i7-6600U - 8Go RAM - Pas de disque dur - Windows - 449.0€ - 14.0'' - 1.6kg - false - false
OMEN 30L GT13-0839nf - Intel Core™ i7-10700K - 32Go RAM - SSD NVMe PCIe WD Black TLC M.2 1 To - Windows - 4499.0€ - NVIDIA GeForce RTX™ 3090
Asus ROG Strix GA35 GA35DX-FR192T - AMD Ryzen 9 - 64Go RAM - SSD libre - Windows - 3999.989990234375€
BUILD SUCCESSFUL (total time: 0 seconds)
```

(Résultat après exécution du Main)