

2. Les collections

- 2.1 ArrayList

```
public class Personne
{
    private String nom;
    private GregorianCalendar dateNaiss;

    public Personne (String n, GregorianCalendar d)
    {
        nom = n;
        dateNaiss = d;
    }

    public String getNom( ) {return nom;}
    public int age( ) {...}

    public String toString( )
    {
        return "La personne dénommée " + nom + " agée de " + age() + " ans";
    }
}
```

```
public class Etudiant extends Personne
```

```
{private int annee;
```

```
private String section;
```

```
public Etudiant (String n, GregorianCalendar d, int an, String s)
```

```
{super(n,d);
```

```
annee = an;
```

```
section = s;}
```

```
public String toString( )
```

```
{return super.toString( ) + " est inscrit en " + annee + "e " + section;}
```

```
}
```

```
import java.util.*;
```

```
public class Principal {
```

```
    public static void main(String[] args)
```

```
{ ArrayList <Personne> arrayPers = new ArrayList <Personne> ( );
```

```
    arrayPers.add(new Personne ("Anne Petit", 12));
```

```
    arrayPers.add(new Personne ("Pierre Leloup", 23));
```

```
    arrayPers.add(new Personne ("Jules Bastin", 88));
```

```
    arrayPers.add(new Etudiant ("Pol Louis", 19,2,"info"));
```

```
    for (int i = 1; i <= arrayPers.size( ); i++)
```

```
        System.out.println("Elément " + i + " : " + arrayPers.get(i-1) );
```

```
    for (int i = 1; i <= arrayPers.size( ); i++)
```

```
        System.out.println("age de la personne numéro " + i + " : "
```

```
            + arrayPers.get(i-1).age( ) );
```

typage



Retourne un objet **Personne**



```
Personne p1 = new Personne ("Pol Castor", 50);
```

```
arrayPers.set(2,p1);
```

```
if (arrayPers.contains(p1))
```

```
    System.out.println("arrayPers contient: " + p1);
```

```
arrayPers.remove(1);
```

```
arrayPers.clear( );
```

```
if (arrayPers.isEmpty( ))
```

```
    System.out.println("arrayPers vide");
```

2. Les collections

- 2.1. ArrayList
- 2.2. Collection d'objets et casting

```
ArrayList<Object> liste= new ArrayList<Object> ( );
```

```
liste.add(new Personne ("Anne Petit", 12));
```

```
liste.add(new Personne ("Pierre Leloup", 23));
```

```
liste.add(new Personne ("Jules Bastin", 88));
```

Polymorphisme:
toString() de Personne

```
for (int i = 1; i<= liste.size( ); i++)
```

```
    System.out.println("Elément " + i + " : " +liste.get(i-1) );
```

```
for (int i = 1; i<= liste.size( ); i++)
```

```
    System.out.println("age de la personne numéro " + i + " : " +
```

```
        liste.get(i-1).age( ) );
```

de type Object

```
    System.out.println("age de la personne numéro " + i + " : " +
```

```
        ( (Personne) (liste.get(i-1))).age( ));
```

CASTING

2. Les collections

- 2.1. ArrayList
- 2.2. Collection d'objets et casting
- 2.3. Boucler sur une collection
 - Itérateur


```
ArrayList <Personne> arrayPers = new ArrayList <Personne> ( );
```

```
arrayPers.add(new Personne ("Pierre Leloup", 23));
```

```
...
```

```
Iterator <Personne> itArrayPers = arrayPers.iterator();
```

```
while (itArrayPers.hasNext() )
```

```
    System.out.println("Elément dans ArrayList arrayPers: " + itArrayPers.next() );
```

```
// ou System.out.println("age: " + itArrayPers .next().age( ) );
```

retourne un objet de type Personne

OK

2. Les collections

- 2.1. ArrayList
- 2.2. Collection d'objets et casting
- 2.3. Boucler sur une collection
 - Itérateur
 - Boucle for sur collection

```
Collection <E> myCollection
```

```
for ( E var : myCollection)  
    { ... var ...  
    }
```

Example:

```
Soit : ArrayList< Book > allBooks
```

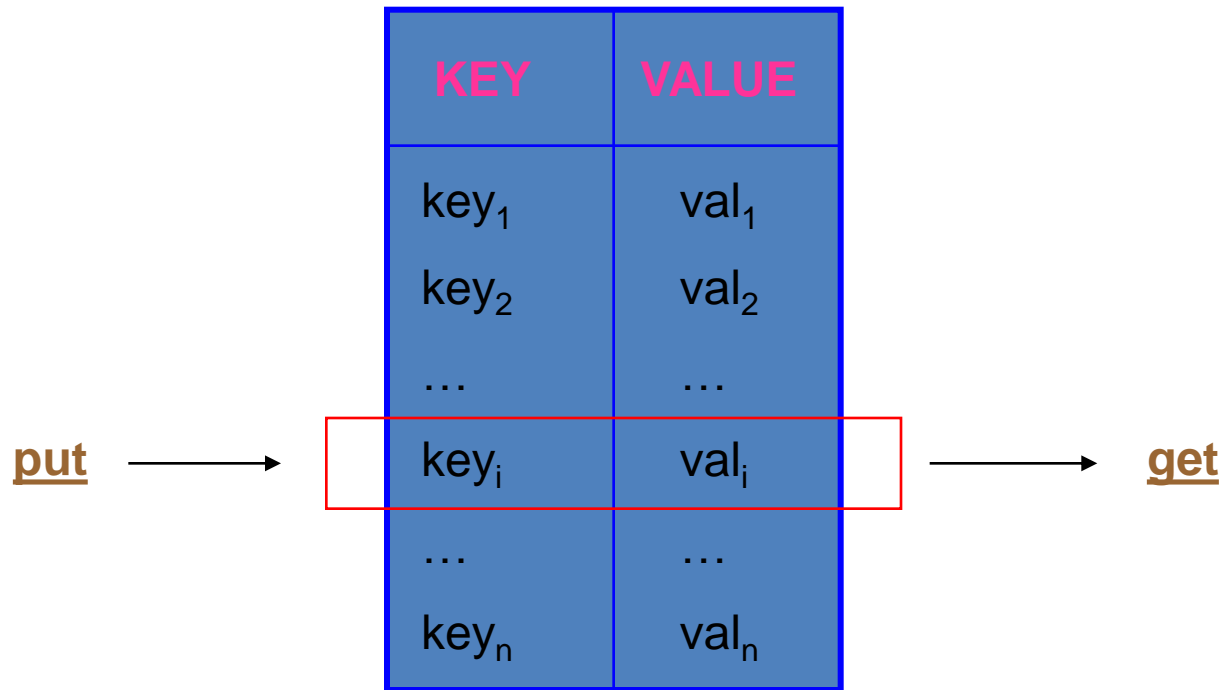
```
for ( Book book : allBooks )
```

```
    System.out.println( book );
```

2. Les collections

- 2.1. ArrayList
- 2.2. Collection d'objets et casting
- 2.3. Boucler sur une collection
- 2.4. HashMap

HashMap



KEY **VALUE**
↑ ↑
HashMap <**String**, **Personne**> **hashPers** = new **HashMap** <**String**, **Personne**> ();

KEYS **VALUES**
↑ ↑
hashPers.put("key1", new **Personne** ("Anne Petit", 8));
hashPers.put("key2", new **Personne** ("Pierre Leloup", 35));
hashPers.put("key3", new **Personne** ("Jules Bastin", 88));

KEY
↑
System.out.println("La valeur correspondant à la clé key1 est: " + **hashPers.get** ("key1"));
System.out.println("La valeur correspondant à la clé key2 est: " + **hashPers.get**("key2"));
System.out.println("La valeur correspondant à la clé key3 est: " + **hashPers.get**("key3"));

```
System.out.println("Nombre de paires clé-valeur dans hashPers est = " + hashPers.size() );
```

```
hashPers.remove("key3");
```

```
if (hashPers.containsKey("key3"))
```

```
    System.out.println("La valeur correspondant à la clé key3 est: " + hashPers.get("key3"));
```

```
else System.out.println("La clé key3 n'existe pas dans hashPers");
```

```
hashPers.clear();
```

```
if (hashPers.isEmpty())
```

```
    System.out.println("hashPers vide");
```


2. Les collections

- 2.1. ArrayList
- 2.2. Collection d'objets et casting
- 2.3. Boucler sur une collection
- 2.4. HashMap
 - Boucler sur une HashMap

```
HashMap <String, Personne> hashPers = new HashMap <String, Personne> ( );
```

```
hashPers.put("key1",new Personne ("Anne Petit", 8) );
```

```
hashPers.put("key2",new Personne ("Pierre Leloup", 35));
```

```
hashPers.put("key3",new Personne ("Jules Bastin", 88));
```

```
for (Personne pers : hashPers.values() )  
    { System.out.println ( "La personne est " + pers ) ; }
```

```
for (String cle : hashPers.keySet() )  
    { System.out.println ( "La clé identifiante est " + cle ); }
```

```
for (Entry<String,Personne> entree: hashPers.entrySet() )  
    { System.out.println ( "La clé identifiante " + entree.getKey() +  
        " correspond à la personne " + entree.getValue() ); }
```

2. Les collections

- 2.1. ArrayList
- 2.2. Collection d'objets et casting
- 2.3. Boucler sur une collection
- 2.4. HashMap
- 2.5. Enumeration

Déclaration:

```
enum EnumName  
    { VALUE1, VALUE2, VALUE3, ...  
    }
```

Utilisation:

```
EnumName.VALUE
```

Exemple:

Déclaration:

```
enum Couleur  
{ ROUGE, BLEU, VERT, JAUNE  
}
```

Utilisation:

```
Couleur.ROUGE
```

Boucler sur les valeurs d'une énumération:

```
for (Couleur couleur : Couleur.values())
```

```
    System.out.println(couleur);
```

