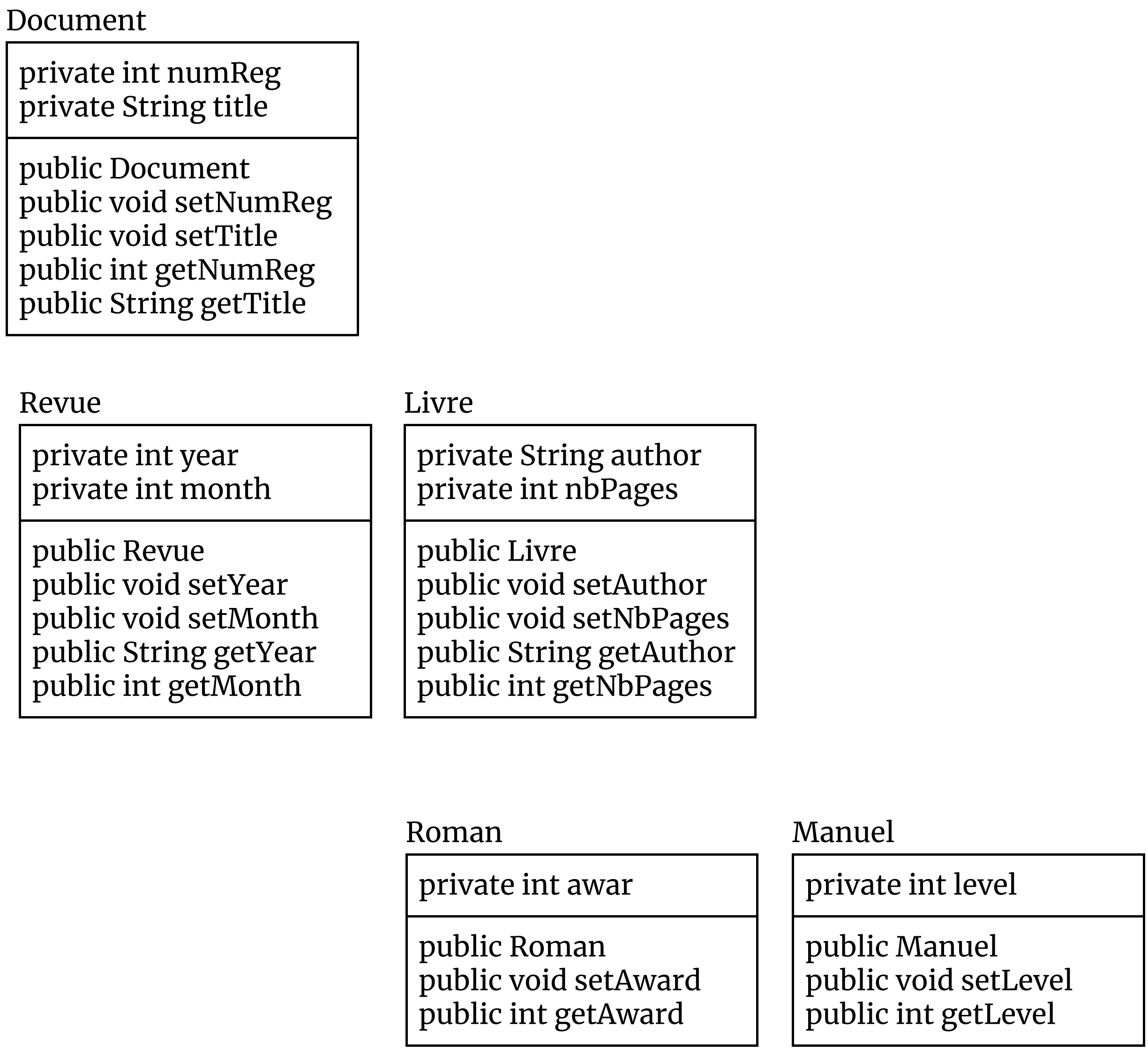
DM2 CS312 Java - Héritage

Bonnet Adrian & Owen Rougé 07/03/2019

A. Diagramme d'héritage



Document

private int numReg
private String title

public Document
public void setNumReg
public void setTitle
public int getNumReg
public String getTitle

Revue

private int year
private int month

public Revue
public void setYear
public void setMonth
public String getYear
public int getMonth

Livre

private String author
private int nbPages

public Livre
public void setAuthor
public void setNbPages
public String getAuthor
public int getNbPages

Roman

private int awar

public Roman
public void setAward
public int getAward

Manuel

private int level

public Manuel
public void setLevel
public int getLevel

B. Classes Document, Livre, Roman, Manuel et Revue

```
public Revue (int num, String tit, int yea, int mon)
{
    super(num, tit);
    this.year = yea;
    this.month = mon;
}

public String toString()
{
    return super(toString()) + "année " + year + ", mois" + month;
}
```

```
public Roman (int num, String tit, String aut, int nb, int awa)
{
    super(num, tit, aut, nb)
    this.award = awa;
}

public String toString()
{
    switch(this.award)
    {
      case 1 :
            return super(toString()) + ", prix Goncourt";
      case 2 :
            return super(toString()) + ", prix Medicis";
      case 3 :
            return super(toString()) + ", prix Interallie";
      default :
            return super(toString());
    }
}
```

C. Classe Bibliothèque

```
public class Bibliotheque
{
    private int volume;
    private List<Document> stock;

    public Bibliotheque (int vol)
    {
      this.volume = vol;
      this.stock = new ArrayList<Document>();
    }

    public Bool add(Document doc)
    {
      Bool ret = false;
      if(stock.size() < volume)
      {
            stock.add(doc);
            ret = true;
      }
      return ret;
    }
    public void printFiles ()
    {
      for (Document doc : stock)
System.out.println(doc.toString());
    }

    public void printAuthors ()
    {
      for (Document doc : stock)      if (doc instanceof Livre)
      System.out.println((Livre)doc.toString());
    }
}
```

D. Méthode ajouterLivres()

```
public int addLivres (bookList)
{
    int occ = 0;
    for(Document doc : bookList)    if(doc instanceof Livre)
    {
      this.add(doc);
      occ ++;
    }
     return occ;
}
```

E. Classe Livrothèque

```
public class Livrotheque
{
    private int volume;
    private List<Livre> stock;

    public Livrotheque (int vol)
    {
      this.volume = vol;
      this.stock = new ArrayList<Livre>();
    }
    public Bool add(Livre doc)
    {
      Bool ret = false;
      if(stock.size() < volume)
      {
           stock.add(doc);
           ret = true;
      }
      return ret;
    }

    public void printFiles ()
    {
      for (Livre doc : stock)
System.out.println(doc.toString());
    }

    public void printAuthors ()
    {
      for (Livre doc : stock)
System.out.println(doc.getAuthor());
    }

    public int addLivres (List<Livre> bookList)
    {
      for(Livre doc : bookList)    this.add(doc);
      return bookList.size();
    }
}
```