Implement the following classes:

```
           AudioBook                              Book
-duration:int                          #title:string
-narrator:string                       #author:string

+AudioBook(ti:string, au:string,       +Book(ti:string, au:string)
          du:int, na:string)           +getTitle():string
+getDuration():int                     +getAuthor():string
+getNarrator():string
```

```
          WrittenBook
#nPages:int

+WrittenBook(ti:string,au:string,
            np:int)
+getNPages():int
```

```
          PrintedBook
-isbn:int                                         EBook
-hardCover:bool                        -size:int

+PrintedBook(isbn:int, ti:string,      +EBook(ti:string,au:string,
            au:string, np:int, hc:bool)          si:int)
+PrintedBook(ti:string,au:string, np:int) +getSize():int
+getIsbn():int                         +print():PrintedBook
+getCoverType():string
```
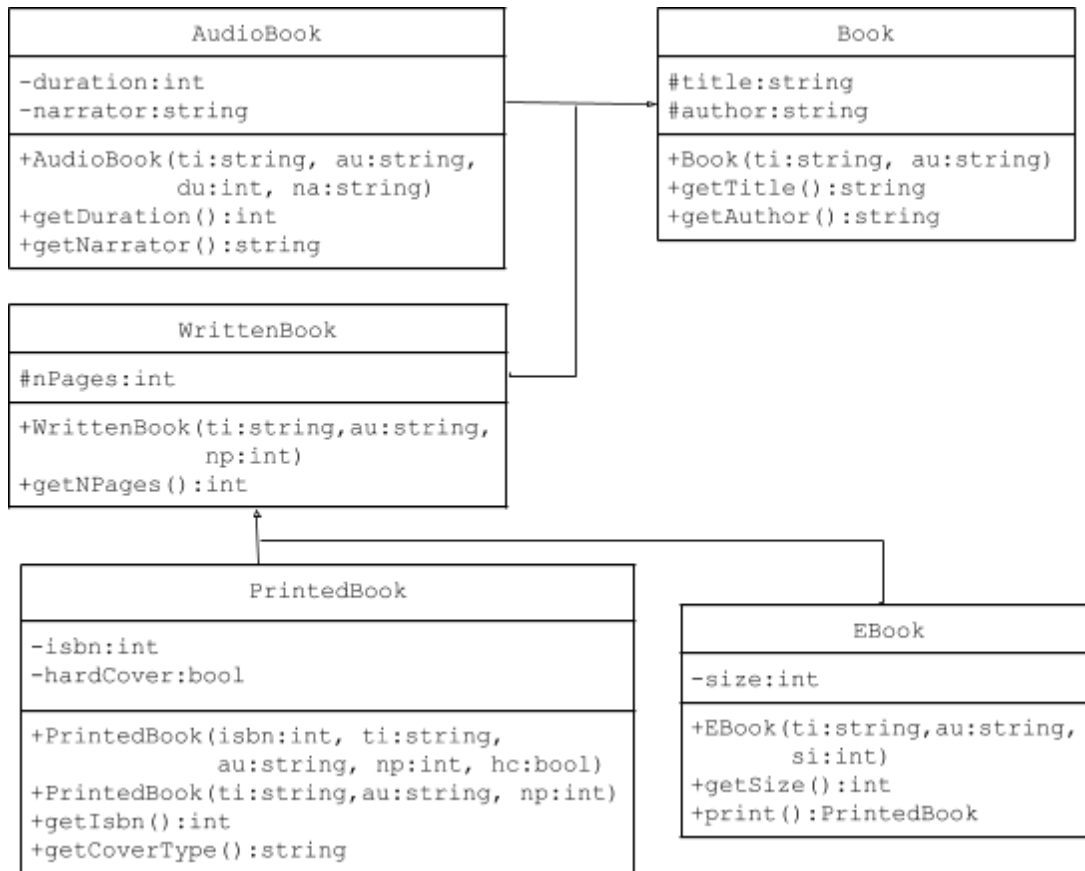
Class Book:
- title and author: represent the title of the book and its author
- Book(ti:string, au:string): a constructor that initializes title and author respectively

Class AudioBook:
- duration: length of the audio clip in minutes
- narrator: name of the person whose voice is recorded
- AudioBook(ti:string, au:string, du:int, na:string): initializes title, author, duration and narrator respectively

Class WrittenBook:
- count of the pages of the book
- WrittenBook(ti:string, au:string, np:int): initializes title, author and nPages respectively

Class PrintedBook:
- isbn: stores the international standard book number
- hardCover: it true if the book has a hardcover and false if it has paperback cover
- PrintedBook(isbn:int, ti:string, au:string, np:int, hc:bool): a constructor that initializes isbn, title, author, nPages and hardCover respectively
- PrintedBook(ti:string, au:string, np:int): a constructor that initializes title, author and nPages to the passed arguments respectively, and initializes isbn to 0 and hardCover to false.

Class EBook:
- size: storage size of the ebook file in MBs
- EBook(ti:string, au:string, si:int): initializes title, author and size respectively
- print(): returns an object of type PrintedBook that has that same title, author and nPages as the current EBook

Solution:

```java
class Book{
  protected String title;
  protected String author;
  public Book(String ti, String au){
    title = ti;
    author = au;
  }
  public String getTitle(){
    return title;
  }
  public String getAuthor(){
    return author;
  }
}
```

```java
class AudioBook extends Book{
  private int duration;
  private String narrator;
  public AudioBook(String ti, String au, int du, String na){
    super(ti, au);
    duration = du;
    narrator = na;
  }
  public int getDuration(){
    return duration;
  }
  public String getNarrator(){
    return narrator;
  }
}
```

```java
class WrittenBook extends Book{
  protected int nPages;
  public WrittenBook(String ti, String au, int np){
    super(ti, au);
    nPages = np;
  }
  public int getNPages(){
    return nPages;
  }
}
```

```java
class PrintedBook extends WrittenBook{
  private int isbn;
  private boolean hardCover;
  public PrintedBook(int isbn,String ti,String au,int np,boolean hc){
    super(ti, au, np);
    this.isbn = isbn;
    hardCover = hc;
  }
```

```java
  public PrintedBook(String ti,String au,int np){
    this(0, ti, au, np, false);
  }
  public int getIsbn(){
    return isbn;
  }
  public String getCoverType(){
    return hardCover? "Hardcover":"Paperback";
  }
}
```

```java
class EBook extends WrittenBook{
  private int size;
  public EBook(String ti, String au, int np, int si){
    super(ti, au, np);
    size = si;
  }
  public int getSize(){
    return size;
  }
  public PrintedBook print(){
    PrintedBook pb;
    pb = new PrintedBook(title, author, nPages);
    return pb;
  }
}
```