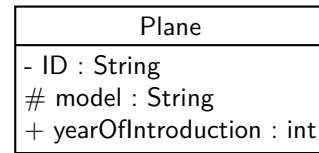
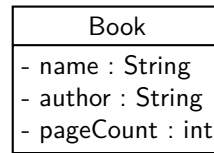
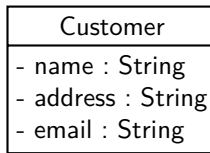


## TD 5 - Génération de code

### Exercice 1. Classes et attributs

Écrire du code java correspondant aux trois classes suivantes.



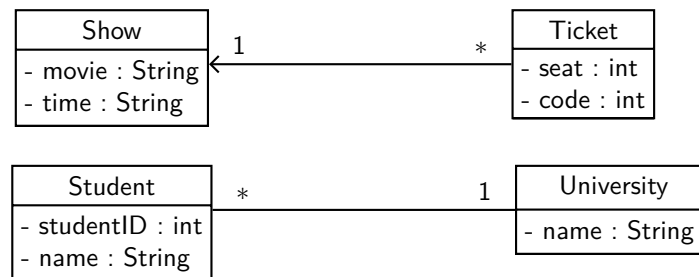
```
1 public class Customer {
2     private String name;
3     private String address;
4     private String email;
5 }
```

```
1 public class Book {
2     private String name;
3     private String author;
4     private int pageCount;
5 }
```

```
1 public class Plane {
2     private String ID;
3     protected String model;
4     public int yearOfIntroduction;
5 }
```

### Exercice 2. Associations

Écrire du code java correspondant aux quatre classes suivantes, ainsi qu'aux associations représentées.



```
1 public class Show {
2     private String movie;
3     private String time;
4 }
```

```
1 public class Ticket {
2     private int seat;
3     private int code;
4     public Show show;
5 }
```

```
1 public class Student {
2     private int studentID;
3     private String name;
4     public University university;
5 }
```

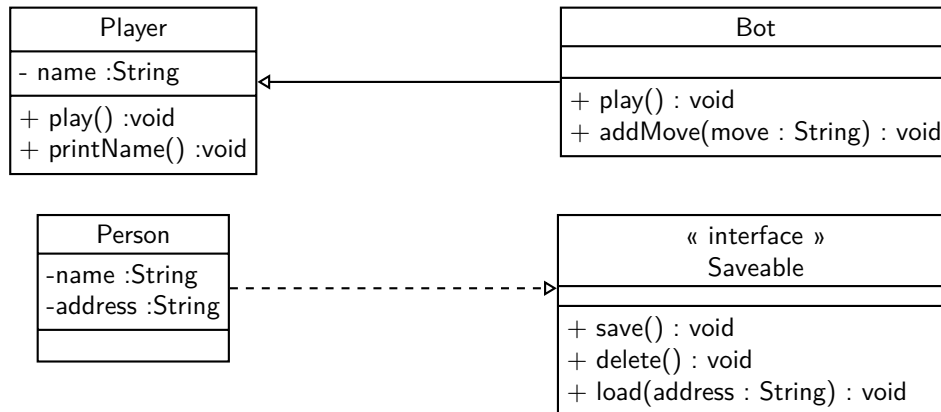
```

1 import java.util.ArrayList;
2
3
4 public class University {
5     private String name;
6     public ArrayList<Student> students;
7 }

```

### Exercice 3. Généralisations et réalisations

Écrire du code java correspondant aux quatre classes suivantes, ainsi qu'aux relations représentées.



```

1 public class Player {
2     private String name;
3
4     public void play() {
5         System.out.println(this.name+ " joue");
6     }
7
8     public void printName() {
9         System.out.println(this.name);
10    }
11 }

```

```

1 public class Bot extends Player {
2     public void play() {}
3     public void addMove(String move) {}
4 }

```

```

1 public interface Saveable {
2     public void save();
3     public void delete();
4     public void load(String address);
5 }

```

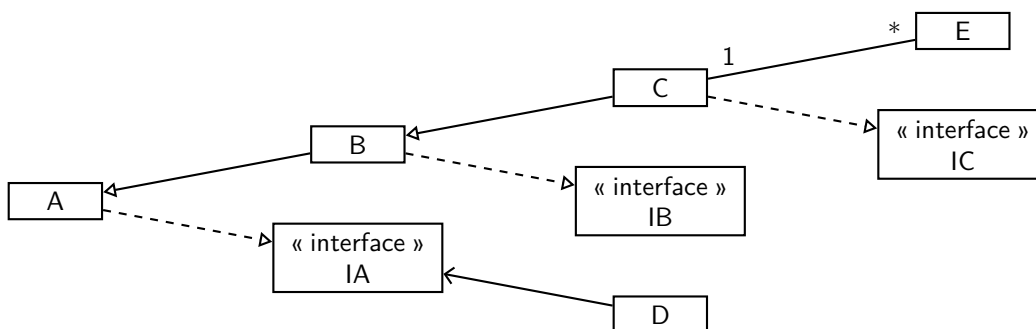
```

1 public class Person implements Saveable{
2     private String name;
3     private String address;
4
5     public void save() {}
6     public void delete() {}
7     public void load(String address) {}
8 }

```

### Exercice 4. Tout à la fois

Écrire du code java correspondant au diagramme suivant.



```
1 public interface IA {
2
3 }
```

```
1 public class A implements IA{
2
3 }
```

```
1 public interface IB {
2
3 }
```

```
1 public class B extends A implements IB{
2
3 }
```

```
1 public interface IC {
2
3 }
```

```
1 import java.util.ArrayList;
2
3 public class C extends B implements IC{
4     public ArrayList<E> es;
5 }
```

```
1 public class D {
2     public IA ia;
3 }
```

```
1 import java.util.ArrayList;
2
3 public class E {
4     public ArrayList<C> cs;
5 }
```