aula06
6.1

Build the Person class which is characterized by name, citizen's card number and date of birth. Start with the following definitions and define new methods to include in the class's public interface.

```
public class Pessoa {
   private String nome;
   private int cc;
   private Data dataNasc;
   // .....
}
```

```
public class Data{
    private int dia;
    private int mes;
    private int ano;
    // .....
}
```

Create suitable methods to allow the initialization of your attributes when creating each object:

```
Data d = new Data(5, 10, 1988);
Pessoa p = new Pessoa("Ana Santos", 98012244, d);
```

Build a new Student class, derived from the Person class, adding the necessary methods and attributes to access and store the mechanographic number (int) and the enrollment date (Date) at the educational institution. Note that the mechanographic number should be assigned automatically (and sequentially from 100) when creating a new student.
The simplified class structure should be as follows:

```
public class Pessoa {
   //...
   String getName(){...}         // retorna o nome da pessoa
}

public class Aluno extends Pessoa {
   //... definição de atributos

   Aluno(String iNome, int iBI, Data iDataNasc, Data iDataInsc);
   Aluno(String iNome, int iBI, Data iDataNasc);
             // nota: neste caso deve assumir a data atual
   int getNMec() {...}          // retorna o número mecanográfico
   // ... acrescentar métodos necessários
}
```

Create the Scholarship class, derived from the Student class, which should include an attribute with the monthly scholarship amount. Define new methods or rewrite the methods you see fit. Add get / set methods associated with the scholarship amount.

Implement the "@Override public String toString ()" method in all classes. For example, for the Person class, it should return:
"Ana Santos, CC: 98012244 Date: 5/10/1988"

Test the work done with the following program:

```
public class Test {
   public static void main(String[] args)  {
   Aluno al = new Aluno ("Andreia Melo", 9855678,
      new Data(18, 7, 1990), new Data (1, 9, 2018));
   Bolseiro bls = new Bolseiro ("Igor Santos", 8976543, new Data(11, 5, 1985));
   bls.setBolsa(1050);

   System.out.println(“Aluno:” + al.getName());
   System.out.println(al);

   System.out.println(“Bolseiro:” + bls.getName() + ", NMec: "
      + bls.getNMec() + ", Bolsa:" +  bls.getBolsa());
   System.out.println(bls);
   }
}
```

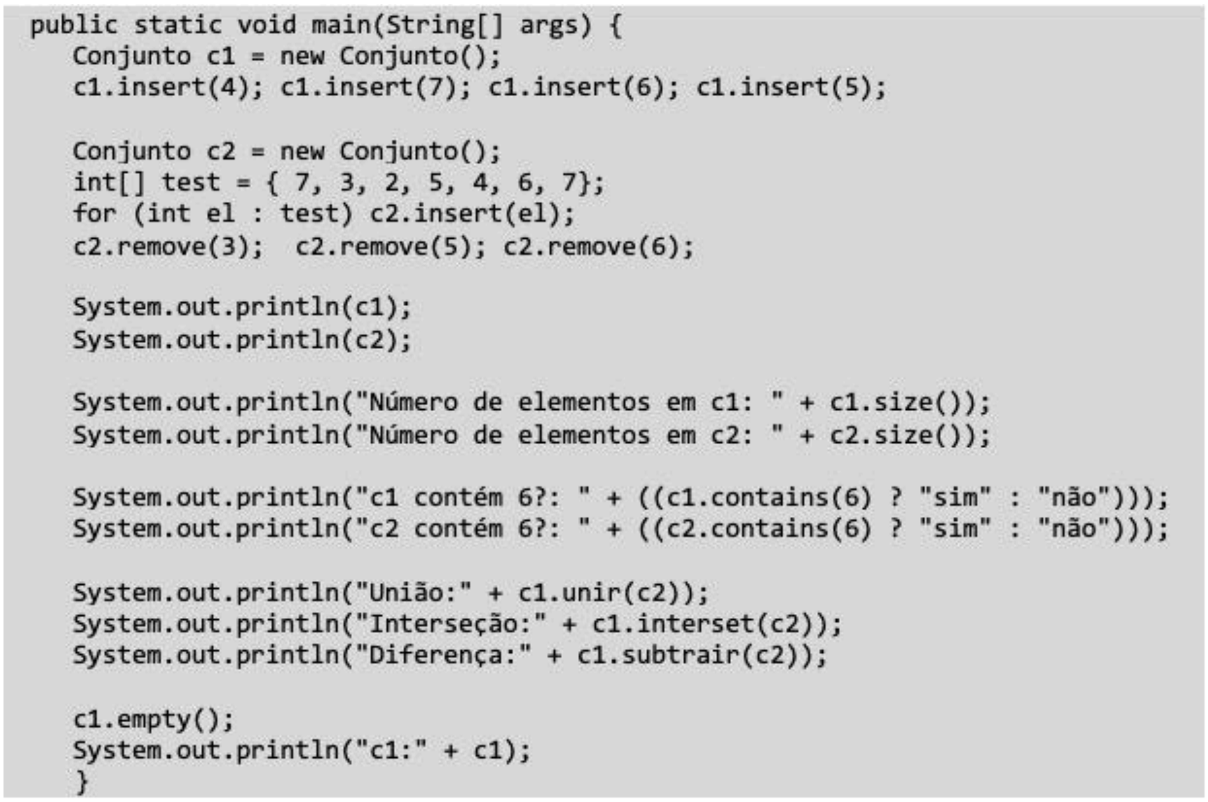
6.2
Using inheritance, rewrite the program developed in class 5 regarding geometric figures. Include a new attribute, color (String), common to all figures.

6.3

Build a Set class that holds a set of integers (which cannot be repeated). Use vectors and implement the following functions:

- void insert (int n); - to insert a new element in the set. If this element already exists, the function does nothing. Initially it is not known how many elements we will insert.
- boolean contains (int n); - to indicate whether a given element is in the set.
- voidremove (intn); - to remove an element from the set. If this element is not in the set, the function does nothing.
- void empty();-para apagar todos os elementos do conjunto.
- String toString (); - to convert the elements of the set into a String.
- int size (); - to calculate the number of elements in the set.
- Set set1 (Set set2); - to build a new set that represents the union of two sets. The resulting set must not contain elements repeated.
- Set subtract (Set diff) - to build a new set that represents the difference of this and the elements of the set represented by diff object.
- Set interset (Set inter) - to build a new set that represents the intersection of this with the elements of the set represented by the inter object. The resulting set cannot contain repeated elements.

Test the developed class with the following main function:



```
public static void main(String[] args) {
   Conjunto c1 = new Conjunto();
   c1.insert(4); c1.insert(7); c1.insert(6); c1.insert(5);

   Conjunto c2 = new Conjunto();
   int[] test = { 7, 3, 2, 5, 4, 6, 7};
   for (int el : test) c2.insert(el);
   c2.remove(3);  c2.remove(5); c2.remove(6);

   System.out.println(c1);
   System.out.println(c2);

   System.out.println("Número de elementos em c1: " + c1.size());
   System.out.println("Número de elementos em c2: " + c2.size());

   System.out.println("c1 contém 6?: " + ((c1.contains(6) ? "sim" : "não")));
   System.out.println("c2 contém 6?: " + ((c2.contains(6) ? "sim" : "não")));

   System.out.println("União:" + c1.unir(c2));
   System.out.println("Interseção:" + c1.interset(c2));
   System.out.println("Diferença:" + c1.subtrair(c2));

   c1.empty();
   System.out.println("c1:" + c1);
   }
```

The results should be as follows (the order of the elements is irrelevant):

```
4 7 6 5
7 2 4
Number of elements in c1: 4 Number of elements in c2: 3 Does c1 contain 6 ?: yes
c2 contains 6 ?: no
Union: 4 7 6 5 2 Intersection: 7 4
Difference: 6 5
c1:
```