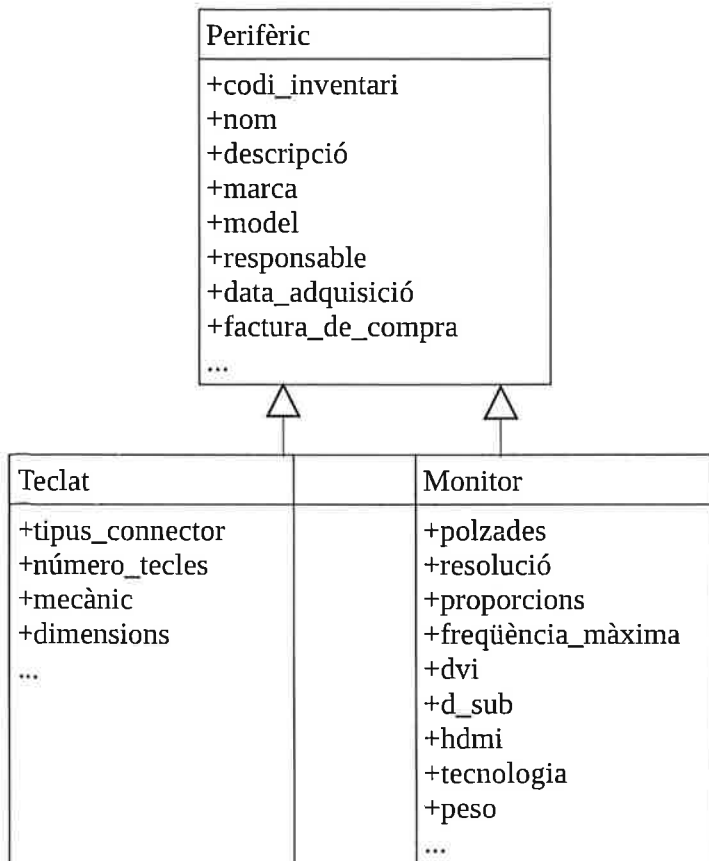


Cas S5

1) En confeccionar el diagrama de classes d'una aplicació de gestió del parc informàtic, ens trobam amb la relació d'herència entre les següents classes:



Es demana transformar el model de classes en el model físic de la base de dades necessària per emmagatzemar els objectes d'aquestes classes, dibuixant el diagrama del model físic de dades i especificant les taules resultants amb les seves columnes corresponents, en cada un dels següents casos d'ús:

- Quan les subclasses tenen molts d'atributs diferents, els tractaments es dirigiran principalment al conjunt total dels perifèrics, especialment els atributs comuns a tots ells i és probable que en el futur apareguin nous tipus de perifèric que interressi tractar. (Valor: 16,66%)
- Quan els tractaments sempre es dirigeixen a tipus específics de perifèrics i interessa estalviar el màxim d'espai possible. (Valor: 16,66%)
- Quan les subclasses es diferencien en pocs atributs, els tractaments sovint es dirigiran al conjunt total dels perifèrics, és important la velocitat d'accés a tota la informació d'un determinat perifèric i no es preveu que apareguin nous tipus de perifèric. (Valor: 16,66%)

Valor de la pregunta: 50% de la nota del cas

2) Explicar quina funció realitzen els següents programes en Java:

a) (Valor: 12,5%)

```
public class X
{
    public static boolean testnumber(int n) {
        int p = n;
        int r = 0;

        while (p != 0) {
            int d = p % 10;
            r = r * 10 + d;
            p = p / 10;
        }
        return n==r;
    }

    public static void main(String[] args) throws Exception {
        int param=Integer.parseInt(args[0]);

        System.out.print("Result: ");
        System.out.println(testnumber(param));
    }
}
```

b) (Valor: 12,5%)

```
public class X {

    public static void fn(String[] a) {

        if (a == null || a.length < 2) {
            return;
        }

        for (int i = 0; i < a.length / 2; i++) {
            String t = a[i];
            a[i] = a[a.length - 1 - i];
            a[a.length - 1 - i] = t;
        }

    }

    public static void main(String[] args) throws Exception {
        System.out.println(String.join(", ", args));
        fn(args);
        System.out.println(String.join(", ", args));
    }
}
```

c) (Valor: 12,5%)

```
import java.io.BufferedReader;
import java.io.BufferedWriter;
import java.io.FileReader;
import java.io.FileWriter;
import java.io.IOException;
import java.util.ArrayList;
import java.util.Collections;

public class X {
    public static void main(String[] args) throws Exception {
        BufferedReader r = null;
        BufferedWriter w = null;
        ArrayList<String> a = new ArrayList<String>();
        r = new BufferedReader(new FileReader(args[0]));
        String c = r.readLine();
        while (c != null) {
            a.add(c);
            c = r.readLine();
        }
        Collections.sort(a);
        w = new BufferedWriter(new FileWriter(args[1]));
        for (String l : a) {
            w.write(l);
            w.newLine();
        }
        r.close();
        w.close();
    }
}
```

d) (Valor: 12,5%)

```
public class X {
    public static void main(String[] args) {
        String[] r = {"M", "CM", "D", "CD", "C", "XC", "L", "XL", "X", "IX", "V", "IV", "I"};
        int[] d = {1000, 900, 500, 400, 100, 90, 50, 40, 10, 9, 5, 4, 1};
        String s = "";
        int n = Integer.parseInt(args[0]);

        if (n >= 1 && n <= 3999) {
            for (int i = 0; i < 13; i++) {
                while(n >= d[i]) {
                    n = n - d[i];
                    s = s + r[i];
                }
            }
            System.out.println(s);
        } else {
            System.out.println("Error");
        }
    }
}
```

Valor de la pregunta: 50% de la nota del cas