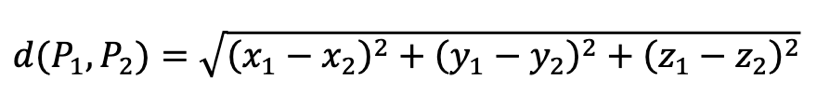
1. Create a class called *Window*. It will have 3 attributes: *name* (string), *width* (int) and *height* (int) and a constructor with 3 parameters to initialize its attributes. It must have as well a method called *toString()* that returns the information of the window. Create a *Main* class where it will have an array of 10 windows. Each *width* and *height* will be assigned using a random number. The *width* will be a number between 90 and 120 cm and the *height* will be a number between 40 and 100 cm.
2. Create a class called *FotoAlbum* with an attribute called *pageNumber*. It will have:
   1. A constructor which receives the page number. (You have to use the word *this* in this constructor).
   2. A constructor without parameters and it will create a *FotoAlbum* with 16 pages. This constructor will reuse the constructor above.
   3. Getter and setter for the attribute.
   4. A *toString* method which returns “I am an album with N pages”.

Create another class called *GranAlbum* that inherits from *FotoAlbum*. This class will have:

1. A constructor without parameters which will reuse the parent constructor and will create an Album with 64 pages.
2. A *toString()* method which will reuse the parent method and will add to the result string “and I am big”.

Create a *Main* class to test both classes. Create an array of 4 albums. 2 will be *FotoAlbum* and the other 2 *GrandAlbum*.

1. Create a class called *3DPoint,*  to represent a point in 3D space, with the coordinates x, y and z as attributes. It has to have these methods:
   1. A *constructor* to stablish the values for x, y and z.
   2. *moveTo(int x, int y, int z)* that will change the coordinates of the point with the new ones received as parameters.
   3. *distanceTo(3DPoint p2)* to calculate the distance to p2 point: (the result will be a double)



* 1. *toString()* method which will return a string with the representation of the point similar to “(2,-7,0)”
  2. All the getters and setters.

You will have to create a *Main* class to test it where you will have to create an array of 5 3D points, ask the user the information about them and after that show the distance between the first one to the others like this:

The distance between (7,4,3) to (17,6,2) is: 10.246951

1. A complex number has 2 parts: a real part and an imaginary part. In a number as a+bi (2-3i, for example) the real part would be “a”(2) and the imaginary part would be “b” (-3). Create a class called *ComplexNumber* to represent complex numbers. It will have:
   1. A constructor to establish both attributes.
   2. A constructor without parameter which reuse the constructor above.
   3. Setters and getters for both attributes.
   4. A *toString*() method that returns “2-3i”
   5. A *getMagnitude*() method that returns the magnitude of the complex number:
   6. A static method *Sum* that returns the sum of 2 complex numbers passed as parameters. The result will be:
2. Create the necessary classes for managing a museum:
   1. We are going to store the information about *Artworks* (abstract class), of each artwork we will store *author*, *owner*, *name* and *year* of creation.
   2. There will be 2 types of artwork: *paintings* and *sculptures*. From the sculptures we will store the material.
   3. We will store the name of the *Author*.
   4. Create a *Main* class to test the classes with a menu that allow the user to introduce the artworks and authors and to show the information about them.
3. Create a class called *House* with:
   1. An attribute *area*.
   2. A constructor to initialize this attribute.
   3. A method called *toString* that returns “I am a house, and my area is X m2”.
   4. The getter and the setter for the attribute area.
   5. A *Door.* Each door will have:
      1. An attribute called *color*
      2. A method *toString* that returns “I am a door, my color is X”.
      3. A getter and a setter for the attribute color.
   6. A getter for the attribute *door.*

Create a class called *SmallApartment* a subclass of *House*, with an area of 50 m2 that will be initialized in the constructor.

Create another class called Person with an attribute *name.* Each *person* will have a *house.* The method *toString* of a person will return its name, the information of its house and the information of the door of the house.

Create a *main* class where you have to create a *smallApartament,* a *person* who lives in it and then show the information of the person.

1. Create an interface called *IMeasurable* with a method called getSize() and another interface called *IDrawable* with a method Draw(). Create an abstract class called *GeometricalFigure* with the attributes x1,y1 (left superior corner), x2,y2 (right inferior corner). You must create 2 classes that inherits from *GeometricalFigure* called Square and Triangle that will implement the 2 interfaces. The class *GeometricalFigure* must have a constructor to initialize all the attributes and another constructor without parameters that will reuse the other constructor. You will have to create also an abstract method *show* that shows if it is a triangle or a square and its size.
   1. Create an abstract class called *Media* with the attributes: author, size in KB, format (for example MPEG4). You must create the getters and setter for the attributes. Create too a constructor to initialize the 3 attributes.
   2. Create a class called *image* that inherits form *Media* and will have 2 more attributes: width and height and their getters and setters. Create its constructor too. It will have to implement the interface IMeasurable that we already did it. The size will be calculated as width x height.
   3. Create a class called *Sound*, it inherits from *Media*, with another 3 attributes: *stereo*(bool), *kbps* and *length* (in seconds), create too their getters and setters and its constructor.
   4. Create a class called *Video* that inherits from *Image*, with another 2 attributes: *codec* and *length*. Create their getters and setters and a constructor.
   5. Finally, you have to create a *Main* class to test the classes. You have to create an array with an object of each type. Show the information of the objects on the console.