PROGRAMACION II

1 – Introducción

Programación II es una asignatura de la carrera Analista Programador. Ésta se dicta durante el primer año en el segundo semestre.
Para aprobar la asignatura hay que tener Programación I.
Para cursar la asignatura se puede deber el Examen Final de Programación I.

2 – Metodología

A continuación se detallan los ejercicios del practico:

2.1. En el ejercicio 1 se pide realizar un código que cree un objeto Auto con constructores, sobrecargas, métodos, getters y setters.

Se realizó una clase llamada Auto, donde se detallan las propiedades a incluir en cada objeto creado en dicha clase, para ellos se crea un constructor de estado inicial donde se le agregan valores por defecto.
Al mismo tiempo se crea una sobrecarga de constructores donde el usuario pueda crear autos y cambiarle las propiedades largo y ancho.
Se crean 4 métodos, 2 getters y 2 setters, en los getters se puede conseguir la información de cada auto creado, mientras que en los setters se permite modificar las otras propiedades de los autos, la de tipo booleano y el tipo de tapicería que el mismo tiene.

2.2. En el ejercicio 2 se pide que se realice un código para poder imprimir en pantalla el nombre de un alumno o el nombre y edad.

Se realizo una clase alumno, la cual contiene un constructor por defecto y una sobrecarga en la cual le podemos pasar parámetros (nombre y edad). Luego dos getters, uno para mostrar el nombre y otro para mostrar la edad y el nombre.
Por último un setter, en el cual podemos modificar su nombre y edad.

2.3. En el ejercicio 3 pide que se cree un código donde se pueda imprimir en pantalla la resta y suma de dos vectores.
Para el mismo se realizo una clase llamada Vector, en esa clase se declararon 6 variables, cuatro de clase array y dos de valor numérico.
Dentro de la clase Vector generamos 4 métodos. Uno de ellos, llamado ingresar(), es el encargado de pedirle al usuario los datos de cada vector, otro, llamado sumaVectores() donde se va a poner la suma de los vectores y al mismo tiempo recorre los vectores a sumar, sumándolos y almacenando su

suma en el primer array. Por último, un método llamado restaVectores(), que funciona de la misma manera que el anterior pero almacena la resta de los dos.

2.4. En el ejercicio 4 se pide que se le agregue al ejercicio 3 un método que pueda calcular la distancia entre los puntos asociados a los vectores creados.

Para eso, dentro de la clase vector se realizo un método el cual calcula la distancia entre los vectores. La misma se calcula utilizando la formula: $d=\sqrt{((x\_2-x\_1)^2+(y\_2-y\_1)^2)}$. Para simplificar primero se realizo la sumas de las potencias y luego a dicho resultado le hallamos su raíz cuadrada.

2.5. En el ejercicio 5 se pide que escribamos un programa para sumar dos matrices.

Se creo una clase que se llama Matriz. Dicha clase recibe por parámetros que el usuario ingresa por consola, para especificar el largo y ancho de las matrices a crear. Una vez recibida las variables se crean dos matrices con números aleatorios y una matriz vacía para utilizar en otros métodos.
En el metodo sumaMatrices() se realiza la suma de dichas matrices y se guardan en la matriz vacia.
En el metodo mostrarMatrices() se muestra la matrizA, matrizB y la suma de las mismas.

2.6. En el ejercicio 6 se pide que se realice un reloj digital. Él mismo deberá contar con la hora, los minutos, los segundos y los milisegundos. Así mismo se pide que se cree un método que permita al usuario cambiar la hora. También que se pueda sumar y restar horas mediante dos métodos distintos.

Se creó la clase Reloj, el cual contiene el funcionamiento del reloj. Esta consta con 2 métodos y 4 atributos.

- aumentar() inicia el reloj desde 00:00:00:00.
- aumentarHora() modifica la hora al pasarle una hora como parámetro. Existe una sobrecarga que trabaja similarmente pasando los minutos y segundos por parámetros.

Por otro lado, se creo la clase Operaciones. La misma crea dos horarios vacíos de clase TimeSpan. Mediante los metodos sumahoras() y restahoras() le pide al usuario que ingrese las horas deseadas a sumar o restar y dandole así el resultado de la operación.

2.7. En el ejercicio 7 se muestra un diagrama de 6 clases con sus métodos y atributos, se pide que se pase al VisualStudio.
En el main se encuentra un switch el cual da un mensaje , el mismo indica las diferentes opciones para ingresar y su respectiva acción.
Por otra parte se realizo una clase padre, que es Impresora y luego muchas clases hijas (factura, remito, facturaluz, municipio y reciboSueldo). La clase impresora ya le da por herencia a cada clase hija un método imprimir. Cada clase hija, a su vez, tiene un método imprimir sobrecargado que imprime

particularmente dentro de su clase. También cada clase hija tiene sus propias propiedades.