

Clase: Programación avanzada

Universidad: Pontificia Universidad Javeriana

Profesor: Juan Sebastián Angarita Torres

TALLER 1. REPASO INTRODUCCIÓN A LA PROGRAMACIÓN

Objetivo: recordar y practicar los temas de programación básica estudiados en la clase de *Introducción a la programación*. Muchos de estos temas van a ser útiles para la clase de *Programación Avanzada* y es importante que los tengan claros para el desarrollo de temas más avanzados.

Recordando los temas que trabajaron en la clase “introducción a la programación” fueron:

- Variables
- Operadores
- Condicionales: if, if/else, switch
- Ciclos: while, do-while, for
- Arreglos y matrices
- Funciones
- Structs

Cada ejercicio valdrá 2.5 unidad. Se calificará según los siguientes parámetros:

- 0.2 si el código está comentado.
- 0.3 si el programa es amigable con el usuario.
- 1.5 si el programa funciona según lo enunciado.
- 0.5 si se siguen buenas prácticas de programación. Estas incluyen:
 - Nombrar variables con cammelCase
 - Toda función debe tener al menos un comentario diciendo que hace
 - Nombre apropiado para las variables
 - Declaración correcta de funciones
 - Código organizado
 - Usar funciones de manera adecuada

Ejercicio 1

Un profesor en la Javeriana está subiendo las notas de sus estudiantes al sistema. Para realizar esta tarea el profesor ingresa cada una de las notas estudiante por estudiante. Es decir, primero ingresa todas las notas del primer estudiante, luego las del segundo estudiante, y así hasta el último.

En total para el semestre se tuvieron 5 notas, cada una calificada entre 0.0 y 5.0, y cada una aporta un porcentaje diferente a la calificación final:

- Primer parcial: 10%
- Segundo parcial: 15%
- Tercer parcial: 25%
- Taller: 5%
- Trabajo en clase: 45%

Su primera tarea es pedir las calificaciones de un estudiante. Posteriormente calcular cual fue la nota final y mostrarla en pantalla. Dependiendo de la calificación final de manera complementaria se debe imprimir un mensaje:

- 0.0 – 0.9: “MUY MAL”
- 1.0 – 1.9: “Toca esforzarse más”
- 2.0 – 2.9: “Casi pasas”
- 3.0 – 3.9: “Felicitaciones”
- 4.0 – 5.0: “Perfecto”

Después de ingresar las notas del primer estudiante se le debe preguntar al docente, si tiene aún mas calificaciones que subir al sistema. En caso de confirmar pide las notas del siguiente estudiante y repite el proceso. En caso de decidir que ya ha calificado a todos los estudiantes se deben realizar las siguientes tareas: organizar las notas finales de mayor a menor, mostrar la mejor y peor calificación, mostrar el promedio de nota final, mostrar cuantos estudiantes perdieron la materia.

Cosas para tener en cuenta:

- Para este ejercicio puede usar todas las instrucciones básicas de programación que conoció en Introducción: if, while, do-while, switch, etc.
- Si la nota ingresada por el profesor no está en un rango de 0.0 a 5.0 debe volver a pedir la calificación hasta que se ingrese de manera correcta.
- No usar structs
- Puede suponer que como máximo hay 50 estudiantes por curso.
- Use funciones para tener un código más organizado y limpio.

Ejemplo

Estudiante 1

- Primer parcial: 5
- Segundo parcial: 4.5
- Tercer parcial: 3.2
- Taller: 4.1

- Trabajo en clase: 2.0

Estudiante 2

- Primer parcial: 2.0
- Segundo parcial: 3.0
- Tercer parcial: 2.0
- Taller: 3.0
- Trabajo en clase: 2.0

Estudiante 3

- Primer parcial: 2.5
- Segundo parcial: 3.5
- Tercer parcial: 2.5
- Taller: 3.5
- Trabajo en clase: 3.0

Notas finales

Estudiante 1: 3.08 – Felicitaciones

Estudiante 2: 2.125 - Casi pasas

Estudiante 3: 2.925 - Casi pasas

Calificaciones Organizadas

3.08

2.925

2.125

Mejor calificación: 3.08

Peor calificación: 2.125

Calificación promedio: 2.71

Estudiantes que perdieron la materia: 2

Ejercicio 2.

Se le ha encargado simular una carrera con los datos de los deportistas que participarán en una competición. Al inicio del programa debe preguntar cuántos deportistas van a participar. Posteriormente pasará a un menú con 4 opciones:

1. Registrar competidor
 2. Mostrar informacion competidores
 3. Mostrar información competidor
 4. Simular carrera
 5. Salir
- Opción 1: debe pedir los siguientes datos del deportista: nombre, edad, peso, altura, velocidad y registrarlo en el sistema.
 - Opción 2: mostrará la información de todos los competidores registrados.
 - Opción 3: pedirá al usuario un número y mostrará la información del competidor que se encuentre en dicha posición del arreglo.
 - Opción 4: simulará la carrera. Para esto todos los corredores comenzarán en la posición 0 y deberán recorrer un total de 200 metros. Ejecute un ciclo donde cada uno de los corredores avanzará tantos metros como indique su atributo velocidad en una iteración. No obstante lo anterior, lo importante para avanzar no es únicamente la velocidad sino la motivación que siente el deportista. Por esto si el corredor va en segundo lugar avanzará su velocidad $\times 1.3$, si va en tercer lugar avanzará su velocidad $\times 1.6$, si va en cuarto lugar avanzará su velocidad $\times 1.9$ y así por cada posición que se aleje del primer lugar tratará de ir 0.3 más rápido. Note que la motivación cambia cada vez que se completa un ciclo, dado que también cambian las posiciones. La velocidad nunca cambia.
- Quando uno de los jugadores complete los 200m, imprima el nombre de todos los jugadores.
- Opción 5: salir del programa.

Ejemplo

Corredor 1

```
{  
    Nombre: pedro  
    Edad: 19
```

```
    Peso: 75
    Altura: 180
    Velocidad: 9
}
```

Corredor 2

```
{
    Nombre: Juan
    Edad: 20
    Peso: 60
    Altura: 165
    Velocidad: 8
}
```

Corredor 3

```
{
    Nombre: Miguel
    Edad: 21
    Peso: 75
    Altura: 180
    Velocidad: 7
}
```

Primera iteración: en la primera iteración dado que todos van empatados simplemente recorren lo que marca su atributo de velocidad.

```
{
    Nombre: pedro
    Distancia recorrida: 9m
}
```

```
{
    Nombre: Juan
    Distancia recorrida: 8m
}
```

```
{
    Nombre: Miguel
    Distancia recorrida: 7m
}
```

Segunda iteración: Lo primero es calcular cuánto avanzada cada uno de los corredores.

- Pedro va en primer lugar así que avanzará únicamente lo que marca su atributo de velocidad es decir 9m.
- Juan va en segundo lugar, su motivación será de $x1.3$. En total avanzará $8m * 1.3 = 10.4m$
- Miguel va en tercer lugar por lo tanto su motivación será de 1.6. En total avanzará $7m * 1.6 = 11.2$

Sumando a lo que ya habían avanzado tenemos

```
{
    Nombre: pedro
    Distancia recorrida: 18m
}

{
    Nombre: Juan
    Distancia recorrida: 18.4m
}

{
    Nombre: Miguel
    Distancia recorrida: 18.2m
}
```

Tercera iteración: Lo primero es calcular cuánto avanzada cada uno de los corredores.

- Pedro va en tercer lugar. Su motivación será de 1.6. En total avanzará $9m * 1.6 = 14.4m$
- Juan va en primer lugar. En total avanzará 8m.
- Miguel va en segundo lugar. Su motivación será de 1.3. En total avanzará $7m * 1.3 = 9.1m$

Sumando a lo que ya habían avanzado tenemos

```
{
    Nombre: pedro
    Distancia recorrida: 32.4m
}

{
    Nombre: Juan
    Distancia recorrida: 26.4m
}
```

{

Nombre: Miguel

Distancia recorrida: 27.3m

}