Prueba 1 – Programación Avanzada

Ingeniería Civil en Computación e Informática – Ingeniería en Tecnologías de Información

8 Mayo 2023 – Duración: 3.5 horas

Nombre		**RUT**		
Paralelo	E.Ross ( ) J.Martínez ( ) J.Rojas ( ) L.Telgie ( )	**Horas de estudio en las últimas 2 semanas**		

Resultados de Aprendizaje

Este instrumento permite evaluar el cumplimiento de los resultados de aprendizaje 1, 2 y 3 del programa de la asignatura:

1. Aplicar técnicas de ingeniería de software en la creación de software legible, mantenible y testeable.

2. Aplicar técnicas de programación orientada al objeto en la resolución de problemas.

3. Comprender la diferencia entre la resolución de problemas utilizando un enfoque procedural versus un enfoque orientado al objeto.

Instrucciones Generales

- **Lea la prueba completamente DOS veces antes de hacer cualquier pregunta.**
- Las entregas pasadas la hora no serán evaluadas y tendrán la nota mínima. Considere que se puede demorar al subir los archivos, así que use el tiempo de entrega para realmente subir los archivos, y no para continuar programando.
- En todos los programas que usted entregue, el archivo que contiene el `main` debe tener su nombre completo como comentario dentro de las primeras 5 líneas. Si el código no contiene el encabezado indicado se descontará un 25% del puntaje obtenido en cada uno de los ejercicios donde no cumplió la instrucción.
- Una prueba respondida correctamente en un 60% corresponde a una nota 4,0.
- Tómese unos minutos para pensar y analizar los problemas antes de comenzar a programar.
- En su espacio personal no debe haber nada más que hojas de papel en blanco, lápiz, goma y/o calculadora. El resto de sus implementos debe guardarlos dentro de su mochila/bolso y ésta debe posicionarse al frente debajo de la pizarra. Si leyó hasta este punto, felicidades, para saber que lo hizo dibuje una estrella al final de esta página.

Prueba 1 – Programación Avanzada

Ingeniería Civil en Computación e Informática – Ingeniería en Tecnologías de Información

8 Mayo 2023 – Duración: 3.5 horas

Problema 1. Ruteo de código (30%)

a) Dado el siguiente código, dibuje y/o describa lo que sucede desde la perspectiva de los objetos que participan

```java
public class Auto {
      private String patente;

      public Auto(String patente) {
         this.patente = patente;
      }
      public String getPatente() {
         return patente;
      }
      public void setPatente(String patente){
         this.patente = patente;
      }
      @Override
      public String toString() {
         return "Auto [patente=" + patente
                                        + "]";
      }
}
```

```java
public class App {

   public static void main
                     (String[] args) {
      Auto a,a1,a2,a3;
      a = new Auto ("p1");
      a1 = a;
      a2 = a1;
      a3 = new Auto("p2");
      a2 = a3;
      a3.setPatente("p1");

      a3.setPatente(a3.getPatente()
                   + a.getPatente());

   }
}
```

b) Dado el siguiente código Java, dibuje y/o describa lo que sucede desde la perspectiva de los objetos que participan y de vector, de manera de mostrar lo que contiene el vector al finalizar el código.

```java
public class Sala {
   private String codigo;

   public Sala(String codigo) {
      this.codigo = codigo;
   }
   public String getCodigo() {
      return codigo;
   }
   public void setCodigo(String codigo) {
      this.codigo = codigo;
   }
   @Override
   public String toString() {
      return "Sala [codigo=" + codigo + "]";
   }
}
```

```java
public class App {
   public static void main(String[]
                                  args){
      Sala [] vector = new Sala[4];
      Sala  s,s1,s2,s3;
      s = new Sala ("c1");
      s1 = s;
      s1.setCodigo("c2");
      s2 = s1;
      s3 = new Sala("c3");
      s2 = s3;
      s2 = new Sala(null);
      vector[0]= s;
      vector[1]= s3;
      vector[2]= s2;
      vector[3]= s1;
   }
}
```

Prueba 1 – Programación Avanzada

Ingeniería Civil en Computación e Informática – Ingeniería en Tecnologías de Información

8 Mayo 2023 – Duración: 3.5 horas

c) Dado el siguiente código Java, indique qué se imprime. Justifique su respuesta

```java
public class App {

   public static void main(
                     String[] args){
      Juego [] vector = new Juego[5];
      Juego j,j1,j2,j3;

      j = new Juego (1,true);
      j3 = j;
      j3.setIndividual(false);
      j2 = j3;
      j3 = new Juego(2,false);
      j1 = j3;
      vector[0]= j;
      vector[1]= j3;
      vector[2]= j2;
      vector[3]= j1;

      for(int i = 0; i < 5; i++) {
         System.out.println(
               vector[i].toString());
      }
   }
}
```

```java
public class Juego {
      private int codigo;
      private boolean individual;

      public Juego(int codigo,
                  boolean individual) {
            this.codigo = codigo;
            this.individual = individual;
      }
      public int getCodigo() {
            return codigo;
      }
      public void setCodigo(int codigo) {
            this.codigo = codigo;
      }
      public boolean isIndividual() {
            return individual;
      }
      public void setIndividual(
                  boolean individual) {
            this.individual = individual;
      }
      public String toString() {
         return "codigo " + codigo +
           ", es individual " + individual;
      }
}
```

Para entregar este problema tiene la opción de subir un archivo a Campus Virtual, o entregarlo de forma física. Si desea entregar la resolución de forma física, indíquelo y se le entregarán hojas en blanco.

Prueba 1 – Programación Avanzada

Ingeniería Civil en Computación e Informática – Ingeniería en Tecnologías de Información

8 Mayo 2023 – Duración: 3.5 horas

Problema 2: Maguito Explosivo (30%)

El querido Maguito Explosivo de 31 minutos cometió un error en su último show de magia y por accidente hizo explotar toda su baraja de cartas. Tras la explosión, se dio cuenta de que no solo la baraja quedó desordenada, sino que también algunas cartas fueron completamente destruidas. Es por esto que le solicita a usted que cree un algoritmo que se encargue de ordenar su baraja.

Una baraja tiene un máximo de 4 pintas y cada pinta tiene un máximo de 13 cartas.

Para esto le entrega un archivo de texto llamado "`baraja.txt`", donde la primera columna muestra la pinta de las cartas **P** (Pica), **C** (Corazón), **T** (Trébol), **D** (Diamantes), seguido de las cartas correspondientes a esa pinta.

```
P,5,A,3,9,J,2,Q,7,8
C,K,J,6,3,9,8,A,4,10,5
D,7,5,9,J,8,K,6,4
T,K,8,A,2,7,4,3,5,9,Q,10,J,6
```

Maguito le solicita que haga un algoritmo que ordene las cartas de la siguiente forma:

- Las **pintas** y sus cartas deben ser ordenadas de forma **descendente**, considerando la cantidad de cartas que poseen las pintas.
- Las filas pares deben ser ordenadas forma **ascendente**: (A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K).
- Las filas impares deben ser ordenadas de forma **descendente**: (K, Q, J, 10, 9, 8, 7, 6, 5, 4, 3, 2, A).

Notas:

- Recuerde y considere que algunas cartas fueron destruidas en la explosión.
- Considere el 0 como un número par.
- Utilice **subprogramas**.
- El ejemplo de archivo de entrada es solo un ejemplo. Las pintas de las cartas pueden venir en cualquier orden.

Debe imprimir por pantalla la baraja final correctamente ordenada.

Por ejemplo:

```
Archivo de entrada:
P,5,A,3,9,J,2,Q,7,8
C,K,J,6,3,9,8,A,4,10,5
D,7,5,9,J,8,K,6,4
T,K,8,A,2,7,4,3,5,9,Q,10,J,6
```

```
Impresión Final:
T,A,2,3,4,5,6,7,8,9,10,J,Q,K
C,K,J,10,9,8,6,5,4,3,A
P,A,2,3,5,7,8,9,J,Q
D,K,J,9,8,7,6,5,4
```

Debe entregar:

- Dibujo de la estructura del programa, que especifique los subprogramas utilizados.
- Código en Java completo.

Prueba 1 – Programación Avanzada

Ingeniería Civil en Computación e Informática – Ingeniería en Tecnologías de Información

8 Mayo 2023 – Duración: 3.5 horas

Problema 3: MUD (40%)

```
11
A1 N A2
A2 O A5
A3 N A8
A2 E A4
A2 N A3
E N A1
A9 E A8
A4 N A6
A10 S A8
A6 N A7
A7 O A8
ENTRADA E
SALIDAS A6 A5
DRAGON A4 A10
ORO A2 A10 A9
N N O O
S S
N N
N N E O
N N N N O E N
N N N N E S
```

Hace algunos años estaban de moda un tipo de juegos denominados MUD (multi-user dungeon, o calabozo multi jugador), en el que una persona se enfrentaba a la tarea de recorrer un calabozo compuesto de muchas habitaciones, y con una serie de desafíos como enemigos y trampas, y recompensas como oro y armas.

En el desafío de hoy, tu misión es recrear un MUD simple, que permita leer desde un archivo tanto la estructura del calabozo como una serie de recorridos que se podrían realizar. La idea final es determinar si alguno de los recorridos logra salir del calabozo.

El archivo tendrá la estructura que se muestra a la derecha. En la primera línea solo existirá un número (**N**), que indica la cantidad de líneas a leer. Después, hay **N** líneas, cada una describiendo una conexión del calabozo. En este ejemplo, si estando en la habitación **A1** se camina al norte se llega a la habitación **A2** (y como las conexiones son bidireccionales, si desde **A2** se camina al sur se llega a **A1**).

Después vendrá una línea que indica cuál es la habitación que se considera la entrada al calabozo. En este ejemplo, es la habitación "**E**". Cuando una persona ingresa al calabozo se considera que se materializa dentro de esa habitación.

Después viene la indicación de qué habitaciones se consideran salidas. Pueden ser muchas, y basta con que la persona llegue a dicha habitación para salir del calabozo.

Después viene una línea indicando en qué habitaciones vive un dragón, y en qué habitaciones hay oro. Si la persona llega a una habitación donde hay oro, se le agrega +1 de oro a su inventario. Si la persona llega a una habitación donde hay un dragón, muere inmediatamente (la persona, no el dragón).

Finalmente, hay una serie de líneas con recorridos que es necesario comprobar. Recuerda que en todos los casos cada recorrido se inicia en la habitación marcada como "entrada" de acuerdo al archivo.

Construya un programa en Java que lea el archivo `mud.txt` y escriba por pantalla el estado de cada recorrido. Por ejemplo, para el archivo anterior se obtendría la siguiente salida por pantalla:

```
En la habitación A2 hay oro.
La habitación A5 es salida.
Saliste del calabozo con 1 de oro.

No existe habitación al S de E

En la habitación A2 hay oro.
Estás de pie en la habitación A2 y no
sabes qué haces.

En la habitación A2 hay oro.
En la habitación A4 hay un dragón. Mueres

En la habitación A2 hay oro.
En la habitación A9 hay oro.
En la habitación A10 hay oro.
En la habitación A10 hay un dragón. Mueres

En la habitación A2 hay oro.
La habitación A6 es salida.
Saliste del calabozo con 1 de oro.
```

Observaciones:

- No hay más de 50 habitaciones en estos MUD.
- Todas las habitaciones tienen potencialmente 4 salidas (N, S, O, E). Si una salida no está configurada, entonces no se puede avanzar en esa dirección.
- Trabaje usando Orientación al Objeto.

Debe entregar:

- Modelo del dominio.
- Código en Java completo.

Prueba 1 – Programación Avanzada

Ingeniería Civil en Computación e Informática – Ingeniería en Tecnologías de Información

8 Mayo 2023 – Duración: 3.5 horas

Con el fin de que al revisar su prueba se entienda mejor lo que realizó, utilice nombres de variables que tengan relación con el propósito que tienen. Si va a crear un contador, prefiera nombres como "contador" o "cont" y evite nombres como "c". Así su profesor no tendrá problemas en interpretar su código. Si leíste esto agrega otra estrella en la primera página.

Compromiso de honestidad

Me comprometo a trabajar con honestidad en esta evaluación, y a que entregaré el resultado de mi trabajo personal.

_____

RUT y Firma

(Debe entregar este documento firmado antes de retirarse. En caso contrario no se revisará su prueba)