

Examen Final

Curso: CC201

Ciclo: 2018.2

Para la preguntas 1 y 2: implemente la clase `a` tal que contenga el método `main` de donde se llamen –iterativamente hasta seleccionar la opción de salida– los métodos de las otras clases. Debe enviar UN SOLO ARCHIVO de nombre `a.java`, no más; para ello, debe completar en los puntos suspensivos:

```
import java.util.Scanner;
// Complete aqui...
public class a {
    public static void main(String[] args){
        Scanner entrada = new Scanner(System.in);
        int opcion;
        do{
            System.out.printf("%n%s%n%s%n%s%n%s%n%n%s",
                "Menu", "0.- salir", "1.- pregunta 1", "2.- pregunta 2",
                "Seleccione una de las opciones: ");
            opcion = entrada.nextInt();
            switch(opcion)
            {
                case 0: break;
                case 1: pregunta1(); break;
                case 2: pregunta2(); break;
            }
        } while(opcion != 0);
    }
    public static void pregunta1(){ // Complete aqui... }
    public static void pregunta2(){ // Complete aqui... }
    // Complete aqui...
}
```

De no seguir las orientaciones de arriba puede ser penalizado hasta con 3 pts por cada pregunta. Debe emplear todas las buenas prácticas de programación vistas en clase: constructores, validación de parámetros, uso de métodos de acceso, uso de `@Override`, etc.

Para la preguntas 3 y 4: desarróllelas en el cuadernillo del examen.

1. (7 pts) Escriba la interfaz llamada `Movable` la cual declara los métodos abstractos `moverse` que al pasarle los puntos inicial y final de un móvil que vive en el plano

cartesiano retorna el recorrido al desplazarse dicho móvil del punto inicial al punto final dados y `obtenerRecorrido` que retorna el recorrido total de dicho móvil. Cada abeja y hormiga están representadas por los objetos `Abeja` y `Hormiga`, respectivamente, que implementan la interfaz `Movable`. Las abejas siguen trayectorias semicircunferenciales, e.g., al pasarle los puntos (0,0) y (2,0), `move` retorna π . Las hormigas siguen trayectorias poligonales, e.g., al pasarle los puntos (0,0) y (3,4), `move` retorna 5. El método `obtenerRecorrido` de `Abeja` retorna el recorrido de la abeja que pasa por los puntos (0,0), (1,2), (2,5) y (4,8); mientras que el de `Hormiga` retorna el recorrido de la hormiga que pasa por los mismos puntos. Cada `Abeja` y `Hormiga` deben tener como atributos su nombre y recorrido total. Tenemos una abeja y una hormiga llamadas maya y fuerte, respectivamente; guarde dichos objetos en el archivo binario `movibles.ser`. Luego, lea dicho archivo y muestre la representación de cadena de cada objeto polimórficamente, utilizando la interfaz `Movable`.

2. (7 pts) Cada libro está representado por un objeto `Libro` que tiene los atributos `titulo`, `autor`, `editorial` y `citaciones`, donde se indica la cantidad de citaciones de dicho documento. Así mismo, cada artículo científico está representado por un objeto `Articulo` que tiene los atributos `titulo`, `autor`, `revista`, donde va el nombre de la revista y las páginas donde fue publicada, y `citaciones`. Cree una clase abstracta `Citable` para que defina `Libro` y `Articulo` empleando herencia. Cree un arreglo que guarde las referencias de los siguientes objetos:

- The C++ programming language. Bjarne Stroustrup. Pearson Education India. [citado por 10304]
- What is object-oriented programming? Bjarne Stroustrup. IEEE software 5 (3), 10-20. [citado por 321]
- Object-oriented database systems: concepts and architectures. Elisa Bertino. Addison-Wesley Longman Publishing Co., Inc. [citado por 341]
- State-of-the-art in privacy preserving data mining. Elisa Bertino. ACM Sigmod Record 33 (1), 50-57. [citado por 956]

A través de este arreglo, usando polimorfismo, escriba la representación de cadena de cada publicación en un archivo de texto `citables.txt`. Finalmente, lea e imprima dicho archivo.

3. (3 pts) Dibuje el diagrama de UML de la pregunta 1.
4. (3 pts) Dibuje el diagrama de UML de la pregunta 2.

12 de diciembre de 2018