Acosta Porcayo Alan Omar, Gutiérrez Grimaldo Alejandro, Medina Villa Samuel

Práctica 5. Abstracción y Encapsulamiento

Resumen

Introducción

Objetivos

-

Metodología

Ejercicio realizado por el profesor

Ejemplo de uso de interfaces.

Código

```
interface Poligono {
  // firma de metodos
  void getArea(int a, int b);
}

class Rectangulo implements Poligono {
  // implementacion del metodo de la interfaz
  public void getArea(int a, int b) {
    System.out.println("El area del rectangulo es: " + (a * b));
  }
}

class Main {
  public static void main(String[] args) {
    Rectangulo r = new Rectangulo();
    r.getArea(4, 8);
  }
}
```

Recreación del juego “Gato”.

```
import java.util.*;

interface Board {
  // En interfaz por lo general los metodos son default
  String checkWinner();
  void printBoard();
}

public class TicTacToe implements Board {
  static String[] board;
  static String player; // Posible objeto que venga de la clase Player L11

  // Override
  public String checkWinner() {
    for(int i = 0; i <8; i ++) {
      String line = null;
      switch(i) {
        case 0: line = board[0] + board[1] + board[2]; break;
        case 1: line = board[3] + board[4] + board[5]; break;
        case 2: line = board[6] + board[7] + board[8]; break;
        case 3: line = board[0] + board[3] + board[6]; break;
        case 4: line = board[1] + board[4] + board[7]; break;
        case 5: line = board[2] + board[5] + board[8]; break;
        case 6: line = board[0] + board[4] + board[8]; break;
        case 7: line = board[2] + board[4] + board[6]; break;
      }

      if(line.equals("XXX")) // Si gana el jugador X
        return "X";
      else if(line.equals("OOO")) // Si gana el jugador O
        return "O";
    }

    for(int a = 0; a < 9; a ++) {
      if(Arrays.asList(board).contains(String.valueOf(a + 1)))
          break;
      else if(a==8)
          return "DRAW";
    }

    System.out.println("\nEs el turneo de " + player + ", ingrese una casilla: ");

    return null;
  }

  //imprimir
  /*
  |-------|-------|-------|
  |   1   |   2   |   3   |
  |-------|-------|-------|
  |   4   |   5   |   6   |
  |-------|-------|-------|
  |   7   |   8   |   9   |
  */

  public void printBoard() {
    System.out.println("\n|---|---|---|");
    System.out.println("| " + board[0] + " | " + board[1] + " | " + board[2] + " |");
    System.out.println("|---|---|---|");
    System.out.println("| " + board[3] + " | " + board[4] + " | " + board[5] + " |");
    System.out.println("|---|---|---|");
    System.out.println("| " + board[6] + " | " + board[7] + " | " + board[8] + " |");
    System.out.println("|---|---|---|");
  }

  public static void main(String[] args) {
    Scanner in = new Scanner(System.in);
    board = new String[9];
    player = "X";
    String winner = null;

    TicTacToe t = new TicTacToe();

    // Llenar la matriz board
    for(int i=0;i<9;i++)
      board[i] = String.valueOf(i+1);

    System.out.println("Bienvenido Tic Tac Toe 3x3");

    t.printBoard();

    System.out.println("Es el turneo de " + player + ", ingrese una casilla: ");
    while(winner == null) {
      int numSlot;

      numSlot = in.nextInt();
      if(!(numSlot>0 && numSlot<=9)) {
        System.out.println("Opcion no valida");
        continue;
      }

      if(board[numSlot-1].equals(String.valueOf(numSlot))) {
        board[numSlot-1] = player; // Player vale "X"
        if(player.equals("X"))
          player = "O";
        else
          player = "X";
        t.printBoard();
        winner = t.checkWinner();
      } else
        System.out.println("El slot ya esta ocupado");
    }
```

Acosta Porcayo Alan Omar Ing. en Computación 320206102
Gutiérrez Grimaldo Alejandro Ing. en Computación 320282098
Medina Villa Samuel Ing. en Computación 320249538

```
    if(winner.equals("DRAW"))
      System.out.println("Nadie gana. Gracias por jugar");
    else
      System.out.println("Ganaste " + winner + " eres un PRO");

    in.close();
  }
}
```

Resultados

Problema 1

Modifique el programa de *Tic Tac Toe* haciendo:

a. El Juego de 5×5 y el que gane lo haga con 4 símbolos unidos.

b. Cree una clase abstracta que sea el jugador (X-O) y herede la clase principal.

Implementar la herencia hibrida, usar interfaces y clases abstractas.

Explicación

Código

Ejecución

Problema 2

Realice un verificador de contraseña usando encapsulamiento de tipo *private* y *protected*. El programa recibe una cadena candidata para ser contraseña y se valida que:

a. Tenga entre 8 y 16 de caracteres

b. Que tenga al menos un carácter especial.

c. Que tenga al menos un número.

d. Que tenga al menos una letra minúscula.

e. Que tenga al menos una letra mayúscula.

Investigue como se hace para ocultar los caracteres escritos en la consola.

Al final regrese si la contraseña es débil, mediana o fuerte.

Implementar encapsulamiento privado y protegido.

Explicación

Código

Ejecución

Problema 3

Programa un ahorcado utilizando la composición de objetos:

a. La palabra adivinanza tiene que ser seleccionada de manera aleatoria de un arreglo de mínimo 30 palabras.

b. El ahorcado deberá mostrarte en consola con una figura simple por ejemplo

   Palabra oculta: "MATEMATICAS"
   Adivina: _ _ _ _ _ _ _ _ _ _ _

c. Si el usuario gana mostrar un mensaje de ganador y de la misma manera si se pierde.

d. Considere las clases:

   i. Palabra (palabras ocultas)

   ii. Ahorcado (figura)

   iii. *Main* (se programa la lógica del juego)

Implementar composición de objetos.

Explicación

Código

Ejecución

Problema 4

El matemático S. Ulam en 1964 planteo una secuencia de números entera que sigue una serie de reglas planteada en https://edabit.com/challenge/RkicZ4kkcSx8K3d4ex. Programe la secuencia de Ulam, pero que reciba desde consola los primeros dos

elementos de la secuencia y se calcule sobre esos dos elementos.

Por ejemplo:

(u, v)	sequence
(1, 2)	1, 2, 3, 4, 6, 8, 11, 13, 16, 18, ...
(1, 3)	1, 3, 4, 5, 6, 8, 10, 12, 17, 21, ...
(1, 4)	1, 4, 5, 6, 7, 8, 10, 16, 18, 19, ...
(1, 5)	1, 5, 6, 7, 8, 9, 10, 12, 20, 22, ...
(2, 3)	2, 3, 5, 7, 8, 9, 13, 14, 18, 19, ...
(2, 4)	2, 4, 6, 8, 12, 16, 22, 26, 32, 36, ...
(2, 5)	2, 5, 7, 9, 11, 12, 13, 15, 19, 23, ...

Para manejar la secuencia haga uso de ArrayList exclusivamente.

Explicación

Código

Ejecución

Conclusiones

Referencias

Solano, J. (2017, 20 enero). *Manual de prácticas de Programación Orientada a Objetos*. Laboratorio de Computación Salas A y B. http://lcp02.fi-b.unam.mx/

Matt. *Ulam Sequence*. Edabit. https://edabit.com/challenge/RkicZ4kkcSx8K3d4e