

## Guía Tarea 3

**Nombre:** Juan David Jiménez Romero

### Clase main:

paquete en el que se encuentra el trabajo

librerías que se usaron para la actividad

inicio de la clase Prueba\_JuanJ\_U2

método que inicializa la aplicación

declaración de las variables y métodos que se usaron en el trabajo como el scan, entre otros

cola para el apartado de cola Prioritaria

Pilas que sirven para almacenar las tareas y tareas completadas

declaración de los valores enteros y string para el almacén de los datos ingresados por el usuario

inicio del menú de opciones que se muestra al abrir el programa

línea para que el usuario ingrese la opción en el menú

while que nos dice que este se repetirá hasta que no sea igual a 0

switch para el menú de opciones

caso 1 que nos sirve para ingresar los datos

se almacena

for que repetirá según las veces que el usuario especifico anteriormente

pide que el usuario ingrese la descripción

almacén la descripción

aquí se cambian los datos de la tarea

se agrega la tarea a la pila de tareas

caso 2 cambiar una tarea como completada

verificación de que este algún valor agregado a la pila

en caso de que si haya datos en la pila se ejecuta lo siguiente

se muestra el valor que está en la cima de la pila

menú que dice que ingrese si quiere marcar sí o no como completada

guarda el valor

while que nos dice que mientras la opción no sea 1 o 2 este se repetirá

menú que dice que ingrese si quiere marcar sí o no como completada

if para que en caso de elegir el 1 este marcara la tarea como completada

se cambia el valor de tarea y se le pone en true

se añade a la pila de tareas completadas

se elimina la tarea de la lista normal

se muestra el ultimo valor agregado a la pila

sí que dice que en caso de que opc sea 2 este mostrara en la pantalla que la tarea no fue realizada

caso para deshacer la última tarea completada

verificación para cuando la pila de tareas completadas este vacía muestre que este esta vacía

en caso de si tener datos mostrara lo siguiente

muestra el ultimo valor de la pila de tareas completadas

menú que dice si el usuario desea deshacer la última tarea

guarda el valor que el usuario ingreso

while que dice que en caso de no ser 1 o 2 este se repetirá

en caso de dar 1 este cambiara a false y se agregara el valor de tarea lista a la pila de tareas por hacer

se añade a la pila de tareas por hacer

se elimina el ultimo valor de la pila de tareas completadas

opción que dice que la tarea no fue deshecha

se muestra el ultimo valor de la pila de tareas por hacer

caso para agregar la última tarea ingresada a la cola de tareas pendientes

verificación del caso 4

añade a la cola prioritaria el último dato ingresado a la pila tareas por hacer  
muestra el valor de la cola

caso 5 que imprime los valores de la pila de tareas no realizadas

verificación del caso 5

caso 6 que imprime los valores de la pila de tareas realizadas

verificación caso 6

for que muestra todos los valores de la pila

caso 7 para ordenar la pila con la librería collections

verificación del caso 7

método de la librería colección para ordenar de forma descendente

caso 8 para mostrar las tareas vencidas

verificación del caso 8

imprime todos los valores vencidos de la cola de prioridad

caso “default” de no seleccionarse ninguno de los valores definidos en el menu

finalización de la clase main.

## Codigo Main:

```
package prueba_juanj_u2;//paquete en el que se encuentra el trabajo
```

```
//librerias que se usaron para la actividad
```

```
import java.util.Collections;
```

```
import java.util.LinkedList;
```

```
import java.util.Queue;
```

```
import java.util.Scanner;
```

```
import java.util.Stack;
```

```
public class Prueba_JuanJ_U2 {//inicio de la clase Prueba_JuanJ_U2
```

```
    public static void main(String[] args) {//metodo que inicializa la aplicacion
```

//declaracion de las variables y metodos que se usaron en el trabajo como el scan,  
entre otros

```
Scanner scan = new Scanner(System.in);
```

```
Scanner scantxt = new Scanner(System.in);
```

```
Tarea tarea = new Tarea(0," ",false);
```

```
//cola para el apartado de colaPrioritaria
```

```
Queue<String> colaPrio = new LinkedList<String>();
```

```
//Pilas que sirven para almacenar las tareas y tareas completadas
```

```
Stack<String> tareasLista = new Stack<String>();
```

```
Stack<String> tareasListacom = new Stack<String>();
```

//declaracion de los valores enteros y string para el almacen de los datos ingresados  
por el usuario

```
int opc,numTarea;
```

```
String a;
```

```
//inicio del menu de opciones que se muestra al abrir el programa
```

```
System.out.println("-----");
```

```
System.out.println("====MENU ARBOL BINARIO====");
```

```
System.out.println("1.Agregar una nueva tarea a la lista de tareas pendientes");
```

```
System.out.println("2.Marcas una tarea como completada");
```

```
System.out.println("3.Deshacer la ultima tarea completada utilizando la pila");
```

```
System.out.println("4.Agregar una tarea a la cola de recordatorios");
```

```
System.out.println("5.Visualizar todas las tareas pendientes");
```

```
System.out.println("6.Visualizar todas las tareas completadas");
```

```
System.out.println("7.Ordenar ");
```

```
System.out.println("8.Visualizar las tareas vencidas de la cola de recordatorios");
```

```
System.out.println("0.Salir");
```

```
System.out.println("-----");
```

```
opc = scan.nextInt();//linea para que el usuario ingrese la opcion en el menu
```

```

while(opc!=0){//while que nos dice que este se repitira hasta que no sea igual a 0

    switch(opc){//switch para el menu de opciones

        case 1://caso 1 que nos sirve para ingresar los datos

            System.out.println("La cantidad de tareas que desea ingresar: ");//se solicita al
            usuario que ingrese la cantidad de datos

            numTarea = scan.nextInt();//se almacena

            for(int i= 0 ; i< numTarea;i++){//for que repitira segun la veces que el usuario
            especifico anteriormente

                System.out.println("Ingrese la descripcion de la tarea "+(i+1)+" : ");//pide
                que el usuario ingrese la descripcion

                a = scantxt.next();//almace la descripcion

                tarea.id = i+1;//aqui se cambian los datos de la tarea

                tarea.desc = a;

                tarea.completada = false;

                tareasLista.add(tarea.getTarea());//se agrega la tarea a la pila de tareas
            }

            break;

        case 2://caso 2 cambiar una tarea como completada

            if(tareasLista.isEmpty()){//verificacion de que este algun valor agregado a la
            pila

                System.out.println("La lista de tareas esta vacia");

            }else{//en caso de que si haya datos en la pila se ejecuta lo siguiente

                System.out.println(tareasLista.peek());//se muestra el valor que esta en la
                cima de la pila

                System.out.println("Desea marcar como completada la tarea? ");//menu que
                dice que ingrese si quiere marcar si o no como completada

                System.out.println("1.SI ");

                System.out.println("2.NO ");

                opc = scan.nextInt();//guarda el valor

```

while(opc!=1 && opc!=2){//while que nos dice que mientras la opcion no sea 1 o 2 este se repitira

System.out.println("ingrese un opcion valida");

System.out.println("Desea marcar como completada la tarea? ");//menu que dice que ingrese si quiere marcar si o no como completada

System.out.println("1.SI ");

System.out.println("2.NO ");

}

if (opc ==1){//if para que en caso de elegir el 1 este marcara la tarea como completada

tarea.completada = true;//se cambia el valor de tarea y se le pone en true

System.out.println("La tarea a sido completada");

tareasListacom.add(tarea.getTarea());//se añade a la pila de tareas completadas

tareasLista.pop();//se elimina la tarea de la lista normal

System.out.println(tareasListacom.peek());//se muestra el ultimo valor agregado a la pila

}

if(opc == 2){//si que dice que en caso de que opc sea 2 este mostrara en la pantalla que la tarea no fue realizada

tarea.completada = false;

System.out.println("La tarea aun no a sido completada");

}

}

break;

case 3://caso para deshacer la ultima tarea completada

if(tareasListacom.isEmpty()){//verificacion para cuando la pila de tareas completadas este vacia muestre que este esta vacia

System.out.println("La lista de tareas completadas esta vacia");

}else{//en caso de si tener datos mostrara lo siguiente

System.out.println(tareasListacom.peek()); //muestra el ultimo valor de la pila de tareas completadas

System.out.println("Desea deshacer la ultima tarea completada? "); //menu que dice si el usuario desea deshacer la ultima tarea

System.out.println("1.SI ");

System.out.println("2.NO ");

opc = scan.nextInt(); //guarda el valor que el usuario ingreso

while(opc!=1 && opc!=2){ //while que dice que en caso de no ser 1 o 2 este se repitira

System.out.println("ingrese un opcion valida");

System.out.println("Desea deshacer la ultima tarea completada? ");

System.out.println("1.SI ");

System.out.println("2.NO ");

opc = scan.nextInt();

}

if (opc ==1){ //en caso de dar 1 este cambiara a false y se agregara el valor de tarea lista a la pila de tareas por hacer

tarea.completada = false;

System.out.println("La tarea a sido deshecha");

tareasLista.add(tarea.getTarea()); //se añade a la pila de tareas por hacer

tareasListacom.pop(); //se elimina el ultimo valor de la pila de tareas completadas

}

if(opc == 2){ //opcion que dice que la tarea no fue deshecha

System.out.println("La tarea no fue deshecha");

}

System.out.println(tareasLista.peek()); //se muestra el ultimo valor de la pila de tareas por hacer

}

```
break;
```

case 4://caso para agregar la ultima tarea ingresada a la cola de tareas pendientes

```
if(tareasLista.isEmpty()){//verificacion
```

```
    System.out.println("La lista de tareas esta vacia");
```

```
}else{
```

```
    System.out.println("Ingresando la tarea a la cola prioritaria");
```

colaPrio.add(tareasLista.pop());//a ade a la cola prioritaria el ultimo dato ingresado a la pila tareas por hacer

```
    System.out.println(colaprio.peek());//muestra el valor de la cola
```

```
    System.out.println("A sido ingresado a la cola prioritaria");
```

```
}
```

```
break;
```

case 5://caso que imprime los valores de la pila de tareas no realizadas

```
if(tareasLista.isEmpty()){//verificacion
```

```
    System.out.println("La Lista de tareas esta vacia");
```

```
}else{
```

pila  
for(int j=0;j<tareasLista.size();j++){//for que muestra todos los valores de la

```
    System.out.println(tareasLista.pop());
```

```
}
```

```
}
```

```
break;
```

case 6://caso que imprime los valores de la pila de tareas realizadas

```
if(tareasListacom.isEmpty()){//verificacion
```

```
    System.out.println("La lista esta vacia");
```

```
}else{
```

de la pila  
for(int k=0;k<tareasListacom.size();k++){//for que muestra todos los valores



```

        System.out.println(tareasListacom.pop());
    }
}
break;
case 7://caso para ordenar la pilas con la libreria colleccions
    if(tareasLista.isEmpty()){//verificacion
        System.out.println("No hay datos en la Lista de tareas");
    }else{
        Collections.sort(tareasLista, Collections.reverseOrder());//metodo de la
libreria coleccion para ordenar de forma descendete
        System.out.println("La lista a sido ordenada");
    }
    break;
case 8://caso para mostrar las tareas vencidas
    if(colaPrio.isEmpty()){//verificacion
        System.out.println("La cola prioritaria esta vacia");
    }else{
        System.out.println("Lista de los trabajos vencidos de la cola de prioridad");
        for(int l=0;l<colaPrio.size();l++){//imprime todos los valores vencidos de la
cola de prioridad
            System.out.println(colaPrio.poll());
        }
    }
    break;
default://caso de no seleccionarse ninguno de los valores definidos en el menu
    System.out.println("Opcion no valida");
    break;
}

```

```

        System.out.println("-----");
        System.out.println("====MENU ARBOL BINARIO====");
        System.out.println("1.Agregar una nueva tarea a la lista de tareas pendientes");
        System.out.println("2.Marcar una tarea como completada");
        System.out.println("3.Deshacer la ultima tarea completada utilizando la pila");
        System.out.println("4.Agregar una tarea a la cola de recordatorios");
        System.out.println("5.Visualizar todas las tareas pendientes");
        System.out.println("6.Visualizar todas las tareas completadas");
        System.out.println("7.Ordenar ");
        System.out.println("8.Visualizar las tareas vencidas de la cola de recordatorios");
        System.out.println("0.Salir");
        System.out.println("-----");
        opc = scan.nextInt();
    }

    System.out.println("Gracias por ingresar");
}
}

```

## Clase Tarea:

clase producto que es la clase que tiene como propósito la función de agregar tareas

declaración de métodos que se utilizaran en la clase producto

constructor de la clase producto donde se inicializan los datos de la clase Tarea

método de tipo string que nos devuelve el producto que se ingrese con un formato en específico que se está dentro de sí.

## Código Tarea:

```
package prueba_juanj_u2;
```

```
//clase producto que es la clase que tiene como propocito la funcion de agregar productos
al restaurante
```

```

import java.util.Stack;

public class Tarea {

    //declaracion de metodos que se utilizaran en la clase producto

    int id;

    String desc;

    boolean completada;

    Stack<Tarea> tareasLista = new Stack<Tarea>();

    //constructor de la clase producto

    public Tarea(int id, String desc, boolean completada){

        this.id = id;

        this.desc = desc;

        this.completada = completada;

    }

    //constructor de tipo string que nos devuelve el producto que se ingrese

    public String getTarea(){

        String Tar = String.format("""

                ID:      %s

                Descripcion: %s

                Completada: %s

                """, id, desc, completada);

        return Tar;

    }

}

```