



UNIVERSIDAD PRIVADA FRANZ TAMAYO
CARRERA INGENIERIA DE SISTEMAS

DEFENSA HITO2 – TAREA FINAL

Alumno: Adriana Contreras Arancibia
Materia: Programacion III
Carrera: Ing. De Sistemas
Paralelo: Progra(1)
Docente: Lic. William R. Barra Paredes
Fecha: 30/03/2020
Github:

<https://github.com/Adriana-Contreras/progra-3/tree/master/proyecto/src>

Cochabamba-Bolivia

Preguntas Teóricas

1. Defina y muestre ejemplos de la clase Scanner.

La clase scanner permite leer valores que entren por el teclado o datos almacenados en un archivo está localizada en el paquete java.util.

Ejemplos:

Método nextInt: Da lectura por teclado de un número entero.

```
int n;  
  
System.out.print("Introduzca un número entero: ");  
  
n = sc.nextInt();
```

Método nextDouble: Da lectura de un número de tipo double:

```
double x;  
  
System.out.print("Introduzca número de tipo double: ");  
  
x = sc.nextDouble();
```

Método next: Da lectura de una cadena

```
String s;  
  
System.out.print("Introduzca texto: ");  
  
s = sc.next();
```

2. Que es la programación orientada a objetos(POO).

La programación orientada a objetos es una manera de programar específica, donde se organiza el código en unidades denominadas clases, de las cuales se crean objetos que se relacionan entre sí para conseguir los objetivos

3.Cuál es la diferencia entre interfaz y herencia.

Herencia: permite compartir automáticamente métodos y datos ente clases, subclases y objetos

Hay dos tipos de herencia: Herencia Simple y Herencia Múltiple, pero java solo permite la herencia simple.

Interfaz: Un interfaz es una lista de acciones que puede llevar a cabo un determinado objeto.

Con el uso de interfaces se puede "simular" la herencia múltiple que Java no soporta.

4. Qué elementos crees que definen a un objeto.

Un objeto se crea instanciando, un objeto tiene como elemento los atributos los cuales almacenan datos.

5. Que es una clases abstracta y muestre un ejemplo.

Una **clase** abstracta es la que declara la existencia de métodos pero no la implementación de dichos métodos, por lo cual una clase abstracta no se puede instanciar.

Ejemplo:

```
public abstract class Figura
{
    private String color;

    public Figura(String color)
    {
        this.color = color;
    }

    public abstract double calcularArea();

    public String getColor()
    {
        return color;
    }
}
```

```
public class Cuadrado extends Figura
{
    private double lado;

    public Cuadrado(String color, double lado)
    {
        super(color);
        this.lado = lado;
    }

    public double calcularArea()
    {
        return lado * lado;
    }
}
```

```
public class Triangulo extends Figura
{
    private double base;
    private double altura;

    public Triangulo(String color, double base, double altura)
    {
        super(color);
        this.base = base;
        this.altura = altura;
    }

    public double calcularArea()
    {
        return (base * altura) / 2;
    }
}
```

```
import java.util.Scanner;

public class PruebaCuadrado
{
    public static void main(String[] args)
    {
        String colorDelCuadrado;
        double ladoDelCuadrado;

        Scanner teclado = new Scanner(System.in);

        System.out.print("Introduzca el color del cuadrado: ");
        colorDelCuadrado = teclado.nextLine();

        System.out.print("Introduzca el lado del cuadrado: ");
        ladoDelCuadrado = teclado.nextDouble();

        Cuadrado cuadrado1 = new Cuadrado(colorDelCuadrado, ladoDelCuadrado);

        System.out.printf("El área del cuadrado %s es: %f", cuadrado1.getColor(),
cuadrado1.calcularArea(););
    }
}
```

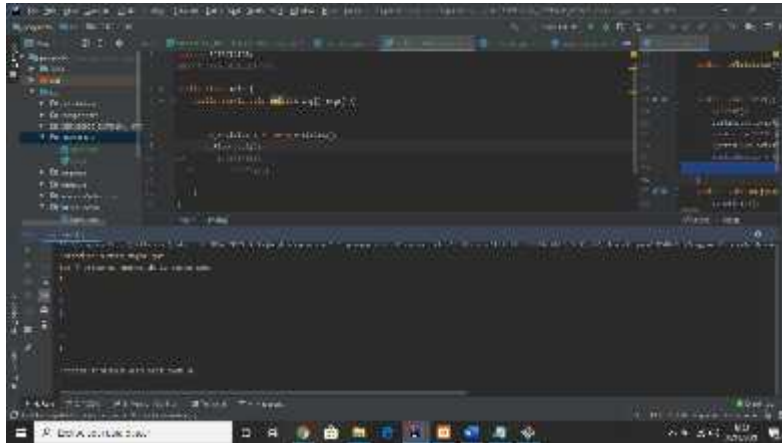
Ejercicio II

Debe de trabajarlos en el package EJERCICIOS.

-) Crear un PACKAGE de nombre EJERCICIOS.
-) Crear una clase de nombre Ejercicios.java.
-) Crear la clase MAIN para mostrar las soluciones.

) Preguntas ejercicios.

- o Generar la serie **fibonacci** hasta un valor n leído por teclado



```
package EJERCICIOS;

import java.util.Scanner;

public class ejercicios {
    Scanner leer = new Scanner(System.in);

    public void Fibonacci() {

        int numero, fi1, fi2, i;
        do {
            System.out.print("Introduce numero mayor que 1: ");
            numero = leer.nextInt();
        } while (numero <= 1);
        System.out.println("Los " + numero + " primeros nmeros de la serie son: ");

        fi1 = 0;
        fi2 = 1;

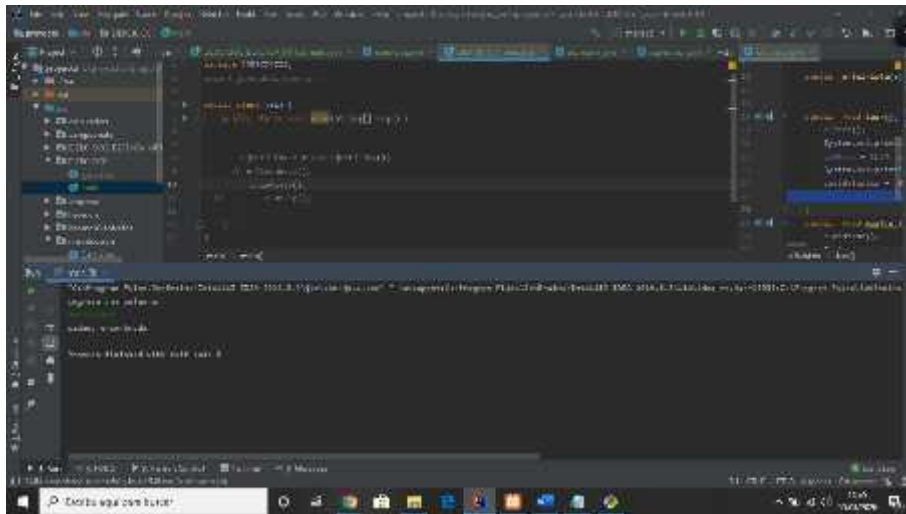
        for (i = 1; i <= numero; i++) {
            System.out.print(fi1);
            fi2 = fi1 + fi2;
            fi1 = fi2 - fi1;

            System.out.println();
        }
    }
}
```

Utilizar la función de Scanner podemos permitir el ingreso por teclado de los valores es por eso que para cada ejercicio instancien una función Scanner propia, en la primera parte controle de que el valor ingresado por el usuario será mayor a 1 para que no te muestre el valor inicial de la serie Fibonacci que sería 0, después estoy imprimiendo los números de la serie fibo asignando los primeros valores por defecto como $fi1 = 0$ y $fi2 = 1$

Usando while y el método **hasNext()** de la clase Scanner, leer N cadenas hasta encontrar una cadena que tenga una cantidad de caracteres igual a 10.*

* Si la cadena ingresada tiene un número igual a 10 caracteres mostrar un mensaje indicando “**Cadena Encontrada**” y salir del while.



```
Scanner leer2 = new Scanner(System.in);

public void cadenas() {

    String palabra;
    int conta = 0;
    System.out.println("ingrese una palabra: ");
    while (leer2.hasNext())
    {
        palabra = leer2.nextLine();

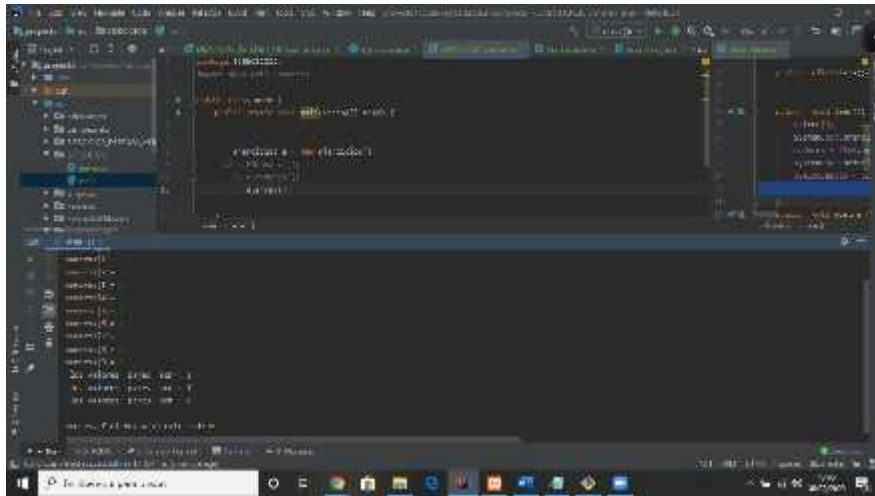
        if (palabra.length() >= 10) {
            conta++;
            System.out.println("cadena encontrada");
            break;
        }

    }

}
```

Aquí hacemos que el usuario ingrese un palabra y mediante el while hacemos que ingrese las veces que sean hasta encontrar una que tenga 10 letras usando la función .length, hacemos el uso tambien del método hasNext que es un método de clase Java escáner que devuelve cierto si hay algo en la entrada y también se bloquea mientras espera la entrada para escanear.

Crear un array con 10 elementos enteros.
Determinar cuántos elementos de ese array son **pares**.



```
Scanner leer3 = new Scanner(System.in);

public void array() {
    int pares = 0;
    int[] num = new int[10];

    System.out.println("Los elementos del array: ");
    for (int i = 0; i < 10; i++) {
        System.out.print("numeros[" + i + "]= ");
        num[i] = leer3.nextInt();
    }
    for (int i = 0; i < 10; i++) {
        if ( num[i] %2 == 0) {
            int contarpar = num[i];
            pares++;
            if (pares != 0) {
                System.out.println(" los valores pares son : " + contarpar );
            } else {
                System.out.println("No ha introducido numeros positivos");
            }
        }
    }
}
```

en este ejercicio nos pide crear un array de 10 posición ingresando los datos mediante teclado, en el cual usamos un primer for para llenar las posiciones del array y un segundo for para contar cuantos números ingresados han sido pares y finalmente lo mandamos a imprimir únicamente pares

Ejercicio III

Esta interface tiene declarado una instancia del objeto Scanner(System.in). Esta interfaz debe ser implementada en la clase Empleado.

El nombre de esta variable es LEER.

Scanner LEER = new Scanner(System.in);

```
package EJERCICIO3_DEFENSA_HIT02;
import java.util.Scanner;

public interface LeerInterfaz {
    Scanner LEER = new Scanner(System.in);
}
```

IV. Manejo de Herencia: Empleado.java

```
1 package EJERCICIO3_DEFENSA_HIT02;
2
3 public class empleado implements LeerInterfaz {
```

V. Manejo de Clases: Gerente.java, Supervisor.java y Oficinista.java

Clase gerente

```
package EJERCICIO3_DEFENSA_HIT02;

public class gerente extends empleado {
    public int bono;
    public int sueldo;
    public int antiguedad;
```

Clase supervisor

```
package EJERCICIO3_DEFENSA_HIT02;

public class supervisor extends empleado{
    public int sueldo;
    public int antiguedad;
    empleado e = new empleado();
```

Clase Oficinista

```
package EJERCICIO3_DEFENSA_HIT02;

public class oficinista extends empleado{
    public String color="a";
    public int sueldobasico;
    empleado e = new empleado();
```


Código

```
package EJERCICIO3_DEFENSA_HIT02;

public class empleado implements LeerInterfaz {
    public String primernombre;
    public String apellido;
    public int ciNumero;
    public String ciExtencion;

    public empleado(){ super();};
    public void leer(){
        System.out.println("ingresar primer nombre: ");
        primernombre = LEER.next();
        System.out.println("ingresar apellido: ");
        apellido = LEER.next();
        System.out.println("ingresar el ci : ");
        ciNumero = LEER.nextInt();
        System.out.println("ingresar la extension del ci: ");
        ciExtencion = LEER.next();
    }
    public void mostrar(){

        System.out.println("primer nombre : " + this.primernombre);
        System.out.println("apellido : " + this.apellido);
        System.out.println("numero de ci: " + this.ciNumero);
        System.out.println("extension de ci: " + this.ciExtencion);
    }
}
```

```

package EJERCICIO3_DEFENSA_HIT02;

public class gerente extends empleado {
    public int bono;
    public int sueldo;
    public int antiguedad;

    empleado e = new empleado();

    //
    // public gerente(String primernombre, String apellido, int ciNumero, String ciExtencion) {
    //     super(primernombre, apellido, ciNumero, ciExtencion);
    //     this.bono = bono;
    //     this.sueldo = sueldo;
    //     this.antiguedad = antiguedad;
    // }

    public gerente(){ super();};
    public void leer(){
        //empleado e = new empleado();
        e.leer();
        System.out.println("ingresar el bono: ");
        bono= LEER.nextInt();
        System.out.println("ingresar sueldo: ");
        sueldo = LEER.nextInt();
        System.out.println("ingresar la antiguedad : ");
        antiguedad = LEER.nextInt();

    }
    public void mostrar(){

        e.mostrar();

        System.out.println("bono: " + this.bono);
        System.out.println("sueldo: " + this.sueldo);
        System.out.println("antiguedad: " + this.sueldo);

    }

}

```

```
package EJERCICIO3_DEFENSA_HIT02;

public class mensajero extends oficinista {
    public String turno;

    oficinista o = new oficinista();

    public mensajero(){ super();};

    public void leer(){
        o.leer();

        System.out.println("ingresar el turno: ");
        turno = LEER.next();

    }
    public void mostrar(){
        o.mostrar();
        System.out.println("turno" + this.turno);

    }
}
```

```

package EJERCICIO3_DEFENSA_HIT02;

public class oficinista extends empleado{
    public String codArea;
    public int sueldoBasico;

    empleado e = new empleado();

    public oficinista(){ super();};

    public void leer(){
        e.leer();
        System.out.println("ingresar el codigo de area: ");
        codArea = LEER.next();
        System.out.println("ingresar el sueldo basico: ");
        sueldoBasico = LEER.nextInt();
    }
    public void mostrar(){
        e.mostrar();
        System.out.println("codigo de area: " + this.codArea);
        System.out.println("sueldo basico : " + this.sueldoBasico);
    }
}

```

```

package EJERCICIO3_DEFENSA_HIT02;

public class secretaria extends oficinista{
    public String nombreArea;

    oficinista o = new oficinista();

    public secretaria(){ super();};

    public void leer(){
        o.leer();
        System.out.println("ingresar el nombre del area : ");
        nombreArea = LEER.next();
    }
    public void mostrar(){

        o.mostrar();
        System.out.println("nombre area: " + this.nombreArea);
    }
}

```

```
package EJERCICIO3_DEFENSA_HITO2;

public class supervisor extends empleado{
    public int sueldo;
    public int antiguedad;
    empleado e = new empleado();

    public supervisor(){ super();};

    public void leer(){
        e.leer();
        System.out.println("ingresar el sueldo : ");
        sueldo= LEER.nextInt();
        System.out.println("ingresar la antiguedad: ");
        antiguedad = LEER.nextInt();

    }
    public void mostrar(){
        e.mostrar();
        System.out.println("sueldo: " + this.sueldo);
        System.out.println("antiguedad: " + this.sueldo);

    }
}
```