

Estructura de datos

15068

Jorge Luis Romo Gonzales

Rodrigo Legarreta Soto

Guía de estudio para primer parcial

23/02/2025

Guía de Estudio: Estructura de Datos

1. ¿Qué es una clase en Java? Explica su importancia en la programación orientada a objetos.

Una clase en Java es como la plantilla de todos los objetos y métodos que se usan en un código, esta sirve como la estructura del programa, aquí se definen atributos y métodos.

2. ¿Qué es un objeto en Java? Proporciona un ejemplo.

Un objeto en Java es algo representado del mundo real en el código, esta es una instancia con valores y atributos que se le da y da acceso a los métodos que uses.

```
public class Person {  
    public String name;  
  
    public static void main(String[] args) {  
        Person objPerson = new Person(); //se crea un objeto  
        objPerson.name = "Rodrigo"; //se le da un atributo a ese objeto  
        System.out.println(objPerson.name); // Se imprime el objeto  
    }  
}
```

3. Explica el significado de public static void main(String[] args) en un programa Java.

Public significa que el programa puede ser accedido desde cualquier parte, static es que se puede ejecutar no importa si hay definido un objeto, void es que no retorna algún valor y no se puede usar return, el main es el punto en el que empieza el programa, sin este java no sabrá como empezar y el código no se ejecutara, el String[] args es los valores en cadena que le puedes dar al código desde la línea de comandos, para que imprima el numero [0, 1, 2]

4. ¿Cómo se declara cada tipo de dato en Java? Da un ejemplo para int, double, String, y boolean.

```
int number = 16;  
double price = 16.99;  
String name = Rodrigo;  
Boolean truefalse = True;
```

5. ¿Cómo se declara un arreglo unidimensional en Java?

Un arreglo unidimensional es una colección de datos, como una lista que le da un valor a cada argumento y esta se escribe así:

```
Int[] numbers = {1, 2, 3, 4, 5}
```

6. Declara un arreglo de enteros y recórrelo con un for tradicional (for i) y con un for each.

```
int[] numbers = {1, 2, 3, 4, 5};

//for tradicional
for (int i = 0; i < numbers.length; i++) {
    { System.out.println(numbers[i]); }
}

//for each
for (int num : numbers){
    { System.out.println(num); }
}
```

7. ¿Cómo se puede determinar si un número es par o impar en Java? Escribe un ejemplo de código.

```
public class ExcersieNonPar {
    public void calculateNonePar() {
        //se crea un arreglo con números para saber cuales tienen que ser
        int[] numeros = {
            0, 2, 5, 8, 11, 14, 17, 20, 23, 26,
            29, 32, 35, 38, 41, 44, 47, 50, 53, 56,
            59, 62, 65, 68, 71, 74, 77, 80, 83, 86,
            89, 92, 95, 98, 0, 3, 6, 9, 12, 15,
            18, 21, 24, 27, 30, 33, 36, 39, 42, 45
        };
    }
}
```

//aqui se crea la variable para que vaya contando los valores que decida el código si son pares nones o cero

```
int even = 0;
```

```
int odd = 0;
```

```
int zero = 0;
```

//se crea un for para que recorra la lista en busca de estos números, si hay un cero en la lista se le agrega al contador de zero, si el numero no deja residuo se le suma al contador de even y todo lo demás se suma al contador de odd

```
for (int i = 0; i < numeros.length; i++) {
```

```
    if (numeros[i] == 0) {
```

```
        zero++;
```

```
    } else if (numeros[i] % 2 == 0) {
```

```
        even++;
```

```
    } else {
```

```
        odd++;
```

```
    }
```

```
}
```

//aquí se imprime el resultado de los contadores y de cuantos números almaceno, imprimirá la cantidad final de los números que haya contado.

```
System.out.println("Cantidad de ceros: " + zero);
```

```
System.out.println("Cantidad de números pares: " + even);
```

```
System.out.println("Cantidad de números impares: " + odd);
```

```
}
```

```
}
```

8. ¿Cómo se declara un arreglo bidimensional en Java?

Se usa la misma logica de un solo arreglo como una lista, pero al ser bidimensional significa que tiene más renglones el arreglo:

```
int[][] matriz = {  
    {1, 2, 3},  
    {4, 5, 6},  
    {7, 8, 9}  
};
```

9. ¿Cómo se puede leer un dato desde la consola en Java? Escribe un ejemplo

se puede leer usando la función scanner ya que esta permite al usuario escribir datos enteros, en cadena o decimales, y una vez utilizado este comando se utiliza un System.out.println para imprimir lo que se haya escrito:

```
import java.util.Scanner;  
  
public class ReadData {  
    public static void main(String[] args) {  
        Scanner scanner = new Scanner(System.in);  
        System.out.print("enter youre name: ");  
        String name = scanner.nextLine();  
        System.out.println("Hola, " + name);  
    }  
}
```

10. ¿Cuál es la nomenclatura recomendada para definir clases en Java?

Usar el UpperCamelCase lo que significa que cada palabra comienza con mayúscula.

11. ¿Cuál es la nomenclatura recomendada para definir funciones (métodos) en Java?

para un método se usa lowerCamelCase esta empieza con una minúscula y sigue con todas las palabras empezando con una mayúscula.

12. ¿Qué es una estructura de datos?

Es una forma de organizar datos, que tengan estructura y así sea más fácil guardar estos mismos, a la hora de tener un código tener esta estructura es fundamental ya que facilita el manejo de los datos y su almacenamiento.

13. ¿Qué es un arreglo y cuáles son sus características?

Un arreglo es una colección de elementos del mismo tipo, como enteros o cadenas de textos que se almacenan en una memoria para que el uso de esa información sea accesible fácilmente.

14. ¿Por qué son importantes las estructuras de datos en la programación?

Estas son muy importantes ya que sin una estructura apropiada los datos almacenados puede que sean complicados de leer o de manejar, si alguien necesita algo rápido u ordenado es mejor tener la información organizada y almacenada de una forma eficaz, esto optimiza el trabajo y lo hace mejor y mas sencillo de trabajar.

15. ¿Qué es un IDE?

Esta significa: Entorno de Desarrollo Integrado, esto quiere decir que es un software que facilita la programación utilizando una biblioteca de herramientas fáciles de usar.

16. ¿Qué IDE utilizas actualmente para programar en Java?

yo utilizo visual studio code para programar, ya que pienso que es una biblioteca amplia, puedo intercambiar entre java o Python y no es específico uno.

17. ¿Cómo puedo saber cuántos elementos tiene un Array en Java?

hay una propiedad específica para eso:

```
int[] nums= {1, 2, 3, 4, 5};  
  
System.out.println(nums.length);
```

Esta función `.length` le da los elementos en la lista y los cuenta

18. ¿Qué es un ArrayList y cuál es la diferencia con un Array normal?

un ArrayList tiene un tamaño dinámico y puede modificarse, un Array tiene un tamaño fijo.

19. ¿Qué es el stack y el heap en la memoria de ejecución de Java?

En el stack se almacenan los datos locales, el último dato es el primero en salir cada que se llama un método se crean nuevos frames de pila que contienen las variables locales del método, por otro lado el heap es una memoria donde se almacenan objetos y sus respectivos atributos, estas se acceden mediante referencias, esto permite compartir datos entre métodos y clases ya que los métodos pueden ser accedidos en cualquier parte del programa

20. ¿Cómo se representa un arreglo en memoria? Explica con un ejemplo.

A cada valor empezando del 0 se le asigna un espacio en la memoria y se representa tal cual.

```
int[] numeros = {5, 10, 15};
```

```
// Memoria: [ 5 | 10 | 15 ]
```

21. Indica si la siguiente afirmación es cierta o falsa : "En Java, todas las sentencias deben terminar con ;".

es falso porque hay estructuras como lo es if, for y while que no usan punto y coma, pero en la mayoría de las sentencias si se usa.

22. Escribe un ejemplo en Java donde se le pasen parámetros a una función y se defina un valor de retorno.

```
public class Sum {  
    public static int plus(int a, int b) {  
        return a + b;  
    }  
    public static void main(String[] args){  
        System.out.println(plus(3, 7));  
    }  
}
```

23. Escribe un ejemplo de como se agregan, se eliminan, se cambian valores, se obtiene su tamaño de un ArrayList

```
import java.util.ArrayList;

public class ArrayListExample {

    ArrayList<String> cars = new ArrayList<>();

    public void addCars(String car) {
        cars.add(car); // Se agrega el dato
    }

    public void set(int index, String car) {
        cars.set(index, car); // Se cambia el valor en el índice
    }

    public void removeCar(int index) {
        cars.remove(index); // Se elimina el dato en el índice
    }

    public int returnSize() {
        return cars.size(); // Retorna el tamaño del ArrayList
    }

    public static void main(String[] args) {
        ArrayListExample objArrayExample = new ArrayListExample();
        objArrayExample.addCars("Mazda"); //se agregan datos
        objArrayExample.addCars("Chevrolet");
    }
}
```

```
        objArrayExample.set(1, "McLaren"); //se editan datos
        objArrayExample.removeCar(0); //remueve el dato

        System.out.println(objArrayExample.returnSize()); // se imprime el
        tamaño de la lista salida: 1
    }
}
```

24. Escribe un ejemplo de interpolación de cadenas en java y uno de concatenación

```
public class UserData {
    public String fullName(String firstName, String lastName) {

        // interpolación
        return String.format("el nombre es: %s %s", firstName, lastName);
        //el .format es para juntar dos variables pero consume mas recursos

        //concatenación
        System.out.println("el nombre es: " + firstName + " apellido: " + lastName);
    }
}
```

Ejercicios

Main:

```
/*
 * guia para examen parcial 1 programas
 * Autor: Rodrigo Legarreta Soto
 * matricula: 15068
 * dia: 23/02/2025
 */
import java.util.Arrays;
import java.util.Scanner;

public class GuiaMain {
    public static void main(String[] args) {
        // 1. Manejo de ArrayList
        System.out.println("Tamano de listas");
        ArrayListManage objArrayExample = new ArrayListManage();
        objArrayExample.addLanguage("Python"); // se agregan datos usando
set
        objArrayExample.addLanguage("Java");
        objArrayExample.addLanguage("C++");
        objArrayExample.addLanguage("SQL");
        objArrayExample.addLanguage("HTML");
        objArrayExample.set(0, "Python actualmente"); // se edita el
lenguaje y al final se pone actualmente
        objArrayExample.removeLanguage(3); // elimino el lenguaje que menos
conozco

        System.out.println(objArrayExample.returnSize()); // imprime el
tamano de la lista que es 4

        objArrayExample.clear(); // borra todo el contenido de la lista

        System.out.println(objArrayExample.returnSize()); // devuelve de
nuevo el tamano que ahora es cero porque fue borrado el contenido

        // 2. Cálculo de ventas con IVA
        System.out.println("Ventas con IVA:"); // se crea un objeto para
llamar a la funcion y que imprima las ventas
        CalculateIva objCalc = new CalculateIva();
        objCalc.calculatePrice();
    }
}
```

```

// 3. Suma de valores en un arreglo bidimensional
System.out.println("Suma total del arreglo:");
ArraySum objSum = new ArraySum();
objSum.bidimensionalSum();

// 4. calculo de calificacion parcial
System.out.println("pedir calificaciones:");

Scanner objScanner = new Scanner(System.in);

System.out.print("Ingresa la calificación del examen: ");
double examScore = objScanner.nextDouble(); // para que el programa
pueda usar decimales ocupamos nextDouble()

System.out.print("Ingresa la calificación de tareas: ");
double homeworkScore = objScanner.nextDouble();

// Llamar al método con valores double
double finalGrade = Grades.calculatePartialGrade(examScore,
homeworkScore);

System.out.printf("Calificación del parcial: %.2f%n", finalGrade);

objScanner.close();

// 5. Conteo de alumnos aprobados y reprobados
System.out.println("calificaciones aleatorias de alumnos:");

int[] grades = FailOrPass.RandomGrades(20);

// aqui se muestran las calificaciones generadas aleatoriamente
System.out.println("Calificaciones de los estudiantes:");
System.out.println(Arrays.toString(grades));

// aqui se llama a la funcion y se crea un objeto
int[] result = FailOrPass.countFailOrPass(grades, 70);

System.out.println("Estudiantes aprobados: " + result[0]);
System.out.println("Estudiantes reprobados: " + result[1]);

}
}

```

1. Manejo de ArrayList

```
/*
 * imprimir lenguajes de programacion
 * Autor: Rodrigo Legarreta Soto
 * matricula: 15068
 * dia: 23/02/2025
 */

import java.util.ArrayList;

public class ArrayListManage {

    ArrayList<String> objLanguages = new ArrayList<>();

    public void addLanguage(String language) {
        objLanguages.add(language); // Se agrega el dato
    }

    public void set(int index, String language) {
        objLanguages.set(index, language); // Se cambia el valor en
el índice
    }

    public void removeLanguage(int index) {
        objLanguages.remove(index); // Se elimina el dato en el
índice
    }

    public int returnSize() {
        return objLanguages.size(); // Retorna el tamaño del
ArrayList
    }

    public void clear() {
        objLanguages.clear(); // elimina todos los datos de la lista
    }

}
```

2. Cálculo de ventas con IVA

```
3.
4. /*
5.  * calcular el iva
6.  * Autor: Rodrigo Legarreta Soto
7.  * matricula: 15068
8.  * día: 23/02/2025
9.  */
10. public class CalculateIva {
11.     public void calculatePrice() {
12.         int[] sales = { 117, 235, 498, 652, 899 }; // Lista de ventas
           del día
13.         double totalSales = 0; // Variable para acumular el total
14.
15.         for (int i = 0; i < sales.length; i++) {
16.             double priceWithIva = sales[i] * 1.16; // se calcula el
           precio con IVA
17.             totalSales += priceWithIva;
18.             System.out.printf("Venta %d: %.2f%n", (i + 1),
           priceWithIva);
19.
20.         }
21.         System.out.printf("Total de ventas con IVA: %.2f%n",
           totalSales);
22.     }
23. }
24.
```

3. Suma de valores en un arreglo bidimensional

```
/*
 * arreglo bidimensional
 * Autor: Rodrigo Legarreta Soto
 * matricula: 15068
 * dia: 23/02/2025
 */
public class ArraySum {
    public void bidimensionalSum() {

        // se inicia el arreglo 3x3
        final int[][] values = {
            { 3, 6, 8 },
            { 16, 9, 17 },
            { 12, 49, 23 },
        };

        int totalSum = 0; // se acumula la suma

        for (int i = 0; i < values.length; i++) {
            for (int j = 0; j < values[i].length; j++) {
                totalSum += values[i][j]; // Se suman los valores
            }
        }

        System.out.println("La suma total de la matriz es: " + totalSum);
    }
}
```


4. Cálculo de calificación parcial

```
/*
 * calcular calificacion del parcial
 * Autor: Rodrigo Legarreta Soto
 * matricula: 15068
 * dia: 23/02/2025
 */
public class Grades {
    public static double calculatePartialGrade(double examScore, double
homeworkScore) {
        return (examScore * 0.7 + homeworkScore * 0.3); // Se usa double
para que soporte decimales
    }
}
```

5. Conteo de alumnos aprobados y reprobados

```
/*
 * calificaciones aleatorias de alumnos
 * Autor: Rodrigo Legarreta Soto
 * matricula: 15068
 * dia: 23/02/2025
 */

import java.util.Random;

public class FailOrPass {

    // se genera un contador para contar aprobados y reprobados
    public static int[] countFailOrPass(int[] grades, int passingGrade) {
        int approved = 0;
        int failed = 0;

        for (int grade : grades) {
            if (grade >= passingGrade) {
                approved++;
            } else {
                failed++;
            }
        }

        return new int[] { approved, failed };
    }

    // aqui se generan los numeros aleatorios entre el 0 y 100
    public static int[] RandomGrades(int numStudents) {
        Random rand = new Random();
        int[] grades = new int[numStudents];

        for (int i = 0; i < grades.length; i++) {
            grades[i] = rand.nextInt(101);
        }

        return grades;
    }
}
```