

Algoritmos y Estructura de Datos – TI

Guía de Problemas - Semana 05

Arreglo lineal: declaración privada e inicialización, recorrido, operaciones básicas y complementarias.

Problema_5_1

Diseñe la clase **ArregloEdades** en el paquete **semana_05** con el atributo privado **edad** (*int*) de tipo arreglo lineal y con los valores de inicialización:

27, 22, 13, 12, 25, 11, 29, 70, 15, 21

Implemente además:

- Un Constructor que no haga nada. ✓
- Un método **tamano** que retorne la cantidad de elementos del arreglo **edad**. ✓
- Un método **obtener** que reciba una posición y retorne la edad registrada en dicha posición. ✓
- Un método **edadPromedio** que retorne el promedio de todas las edades. ✓
- Un método **edadMayor** que retorne la mayor de todas las edades. ✓
- Un método **edadMenor** que retorne la menor de todas las edades. ✓
- Un método **cantMayoresEdad** que retorne la cantidad de personas mayores de edad. ✓
- Un método **cantMenoresEdad** que retorne la cantidad de personas menores de edad. ✓
- Un método **posPrimeraEdadAdolescente** que retorne la posición de la primera edad encontrada en el rango de 12 a 20 años. En caso no exista retorne -1. ✓
- Un método **posUltimaEdadAdolescente** que retorne la posición de la última edad encontrada en el rango de 12 a 20 años. En caso no exista retorne -1. ✓
- Un método **generarEdades** que remplace las edades actuales por otras aleatorias comprendidas en el rango de 10 a 90 años. Haga uso del método: ✓

```
private int aleatorio(int min, int max) {  
    return (int)((max - min + 1) * Math.random()) + min;  
}
```

En la clase principal:

- Declare y cree el objeto global **ae** de tipo **ArregloEdades**. ✓
- A la pulsación del botón **Listar** visualice las edades del arreglo. ✓
- A la pulsación del botón **Reportar** visualice: cantidad de edades, edad promedio, edad mayor, edad menor, cantidad de personas mayores de edad, cantidad de personas menores de edad, posición de la primera edad adolescente y posición de la última edad adolescente. ✓
- A la pulsación del botón **Generar** invoque al método **generarEdades**.

Problema_5_2

Diseñe la clase **ArregloSueldos** en el paquete **semana_05** con el atributo privado **sueldo** (*double*) de tipo arreglo lineal y con los valores de inicialización:

2400.5, 800.4, 1500.2, 1000.3, 4700.1, 600.0, 3300.8, 2600.6, 5100.9, 2000.7

Implemente además:

- Un Constructor que no haga nada. ✓
- Un método **tamano** que retorne la cantidad de elementos del arreglo **sueldo**. ✓
- Un método **obtener** que reciba una posición y retorne el sueldo registrado en dicha posición. ✓
- Un método **sueldoPromedio** que retorne el promedio de todos los sueldos. ✓
- Un método **sueldoMayor** que retorne el mayor de todos los sueldos. ✓
- Un método **sueldoMenor** que retorne el menor de todos los sueldos. ✓
- Un método **cantMayoresSueldoPromedio** que retorne la cantidad de empleados cuyo sueldo es mayor o igual al sueldo promedio. ✓

- Un método **cantMenoresSueldoPromedio** que retorne la cantidad de empleados cuyo sueldo es menor al sueldo promedio.
- Un método **posSegundoSueldoMayorAlMinimo** que retorne la posición del segundo sueldo mayor a 850.0. En caso no exista retorne -1.
- Un método **posUltimoSueldoMenorAlMinimo** que retorne la posición del último sueldo menor a 850.0. En caso no exista retorne -1.
- Un método **generarSueldos** que remplace los sueldos actuales por otros aleatorios comprendidos en el rango de 799.9 a 4999.9 (con una cifra decimal).

En la clase principal:

- Declare y cree el objeto global **as** de tipo ArregloSueldos.
- A la pulsación del botón **Listar** visualice los sueldos del arreglo.
- A la pulsación del botón **Reportar** visualice: cantidad de sueldos, sueldo promedio, sueldo mayor, sueldo menor, cantidad de empleados cuyo sueldo es mayor o igual al sueldo promedio, cantidad de empleados cuyo sueldo es menor al sueldo promedio, posición del segundo sueldo mayor a 850.0 y posición del último sueldo menor a 850.0.
- A la pulsación del botón **Generar** invoque al método generarSueldos.

Propuesto_5_1

Diseñe la clase **ArregloNotas** en el paquete **semana_05** con el atributo privado **nota** (*int*) de tipo arreglo lineal y con los valores de inicialización:

11, 10, 16, 18, 15, 13, 20, 12, 19, 17

Implemente además:

- Un Constructor que no haga nada.
- Un método **tamano** que retorne la cantidad de elementos del arreglo **nota**.
- Un método **obtener** que reciba una posición y retorne la nota registrada en dicha posición.
- Un método **notaPromedio** que retorne el promedio de todas las notas.
- Un método **notaMayor** que retorne la mayor de todas las notas.
- Un método **notaMenor** que retorne la menor de todas las notas.
- Un método **cantNotasAprobatorias** que retorne la cantidad de alumnos que obtuvieron de 13 a más.
- Un método **cantNotasDesaprobatorias** que retorne la cantidad de alumnos que obtuvieron menos de 13.
- Un método **cantNotasMayoresA15** que retorne la cantidad de alumnos que obtuvieron más de 15.
- Un método **posPrimeraNotaAprobatoria** que retorne la posición de la primera nota mayor o igual a 13. En caso no exista retorne -1.
- Un método **posPenultimaNotaDesaprobatoria** que retorne la posición de la penúltima nota menor a 13. En caso no exista retorne -1.
- Un método **generarNotas** que remplace las notas actuales por otras aleatorias comprendidas en el rango de 0 a 20.

En la clase principal:

- Declare y cree el objeto global **an** de tipo ArregloNotas.
- A la pulsación del botón **Listar** visualice las notas del arreglo.
- A la pulsación del botón **Reportar** visualice: cantidad de notas, nota promedio, nota mayor, nota menor, cantidad de notas aprobatorias, cantidad de notas desaprobatorias, cantidad de notas mayores a 15, posición de la primera nota mayor o igual a 13 y posición de la penúltima nota menor a 13.
- A la pulsación del botón **Generar** invoque al método generarNotas.

Propuesto_5_2

Diseñe la clase **ArregloPrecios** en el paquete **semana_05** con el atributo privado **precio** (*double*) de tipo arreglo lineal y con los valores de inicialización:

240.5, 350.2, 80.4, 100.3, 470.1, 60.0, 330.8, 260.6, 510.9, 200.7

Implemente además:

- Un Constructor que no haga nada. ✓
- Un método **tamano** que retorne la cantidad de elementos del arreglo **precio**. ✓
- Un método **obtener** que reciba una posición y retorne el precio registrado en dicha posición. ✓
- Un método **precioPromedio** que retorne el promedio de todos los precios. ✓
- Un método **precioMayor** que retorne el mayor de todos los precios. ✓
- Un método **precioMenor** que retorne el menor de todos los precios. ✓
- Un método **cantMayoresPrecioPromedio** que retorne la cantidad de productos cuyo precio es mayor o igual al precio promedio. ✓
- Un método **cantMenoresPrecioPromedio** que retorne la cantidad de productos cuyo precio es menor al precio promedio. ✓
- Un método **posPrimerPrecioMayorAlSegundo** que retorne la posición del primer precio mayor al segundo de los precios. En caso no exista retorne -1. ✓
- Un método **posUltimoPrecioMenorAlPenultimo** que retorne la posición del último precio menor al penúltimo de los precios. En caso no exista retorne -1. ✓
- Un método **generarPrecios** que remplace los precios actuales por otros aleatorios comprendidos en el rango de 99.9 a 999.9 (con una cifra decimal). ✓

En la clase principal:

- Declare y cree el objeto global **ap** de tipo **ArregloPrecios**. ✓
- A la pulsación del botón **Listar** visualice los precios del arreglo. ✓
- A la pulsación del botón **Reportar** visualice: cantidad de precios, precio promedio, precio mayor, precio menor, cantidad de productos cuyo precio es mayor o igual al precio promedio, cantidad de productos cuyo precio es menor al precio promedio, posición del primer precio mayor al segundo de los precios y posición del último precio menor al penúltimo de los precios. ✓
- A la pulsación del botón **Generar** invoque al método **generarPrecios**. ✓

"Para aceptar el cambio no debemos esperar a que el mundo lo haga primero."

MP