



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Dra. Rocío Alejandra Aldeco Pérez

Asignatura: Programación orientada a objetos -1323

Grupo: 6

No de Práctica(s): 3

Integrante(s): Mendoza Hernández Carlos Emiliano

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada:

Semestre: 2023-1

Fecha de entrega: 9 de septiembre del 2022

Observaciones:

CALIFICACIÓN: _____



Práctica 3.

Utilerías y clases de uso general.

OBJETIVO

- Utilizar bibliotecas propias del lenguaje para realizar algunas tareas comunes y recurrentes.

ACTIVIDADES

- Conocer las bibliotecas del lenguaje.
- Utilizar algunas clases propias de la biblioteca del lenguaje.

INTRODUCCIÓN

Al trabajar en un problema de programación, normalmente se debe verificar si hay clases pre-construidas que satisfagan las necesidades del programa. Si existen esas clases, entonces hay que utilizarlas: ***“no tratar de reinventar la rueda”***.

Hay dos ventajas principales de usar clases pre-construidas: se puede ahorrar tiempo ya que no es necesario escribir nuevas; y el uso de clases pre-construidas también puede mejorar la calidad de los programas ya que han sido probadas completamente, depuradas y sometidas a un proceso de escrutinio para asegurar su eficiencia.



Realiza las siguientes actividades después de leer y revisar en clase la **Práctica de Estudio 3: Utilerías y clases de uso general.**

1. Crea una clase llamada **MiArreglo** que contenga una lista dinámica de tipo **String** usando la clase **ArrayList**.

```
MiArreglo.java > MiArreglo
1  import java.util.ArrayList;
2  import java.util.Collections;
3
4  public class MiArreglo {
5      private ArrayList<String> palabras;
6
7      public MiArreglo(ArrayList<String> palabras) {
8          this.palabras = palabras;
9      }
10 }
```

← *ArrayList de tipo String como único atributo*

← *Constructor que inicializa una instancia de MiArreglo con un ArrayList*

2. Esta clase deberá tener los siguientes métodos:

- **Tamaño:** Regresa el número de elementos de la lista
- **Ordenar:** Ordena la lista de manera decreciente.
- **BúsquedaMayor:** Busca la primera palabra más larga (con mayor número de caracteres) dentro de la lista y regresa el índice de esta.

```
MiArreglo.java > MiArreglo
11  /*Metodos */
12  public int tamano() {
13      return this.palabras.size();
14  }
15
16  public String ordenar() {
17      String ordenadas = "";
18      Collections.sort(this.palabras);
19      for (int i = 0; i < this.palabras.size(); i++) {
20          ordenadas += this.palabras.get(i) + " ";
21      }
22      return ordenadas;
23  }
24
25  public int busquedaMayor() {
26      int index = -1;
27      int aux = 0;
28      for (int i = 0; i < this.palabras.size(); i++) {
29          if ((this.palabras.get(i)).length() > aux) {
30              aux = (this.palabras.get(i)).length();
31              index = i;
32          }
33      }
34      return index;
35  }
36
37 }
```

Usando el método `size()` de `ArrayList`

Usando el método `sort()` de `Collections`. Después, se concatenan todas las palabras ordenadas en una sola cadena, que es regresada por el método.

Recorre todos los `Strings` del `ArrayList` "palabras", y con una variable auxiliar se guarda el tamaño de la palabra mayor después de cada comparación. Al final, devuelve su respectivo índice al terminar del recorrido.



PROGRAMACIÓN ORIENTADA A OBJETOS



3. Crea tu clase principal (Main) leyendo los elementos tipo String dados por Alphagrader. Llama tus tres métodos e imprime los resultados en el orden indicado en el punto 2.

```
1  import java.util.ArrayList;
2  import java.util.Scanner;
3
4  public class Main {
5      Run | Debug
6      public static void main(String[] args) {
7          Scanner stdin = new Scanner(System.in);
8          ArrayList<String> arreglo = new ArrayList<String>();
9          while (stdin.hasNextLine()) {
10             arreglo.add(stdin.nextLine());
11         }
12         stdin.close();
13
14         MiArreglo unArreglo = new MiArreglo(arreglo);
15         System.out.println("Tamano: " + unArreglo.tamano());
16         System.out.println("Lista ordenada: " + unArreglo.ordenar());
17         System.out.println("Indice palabra mayor: " + unArreglo.busquedaMayor());
18     }
19 }
```

Creando un *ArrayList* con datos tipo *String*

Leyendo el *input* en "*arreglo*"

Instanciando la clase *MiArreglo*.
Parámetro: "*arreglo*" (*ArrayList* con las palabras del input)

Llamando a los métodos de *unArreglo*



CONCLUSIONES

En esta práctica se desarrolló un programa que permitió la implementación y comprensión de los siguientes conceptos:

- Revisión de la API de Java con sus clases y sus respectivos métodos, gracias a ello fue posible ordenar el ArrayList con el método Collections.sort().
- Uso y manejo de arreglos dinámicos de tipo String.
- Manejo y concatenación de Strings, para dar formato a la salida del programa.

El uso correcto de las bibliotecas y clases pre-construidas puede facilitar el trabajo, además de que mejora la calidad y tiene un funcionamiento más claro dado que ya han sido probadas completamente, depuradas y sometidas a un proceso de escrutinio para asegurar su eficiencia.

Considero que se cubrieron todos los objetivos de la práctica, dado que se investigó y revisó la API de JAVA con el fin de buscar las clases que más facilitaran el programa.



REFERENCIAS

- *Martín, Antonio*
Programador Certificado Java 2.
Segunda Edición.
México
Alfaomega Grupo Editor, 2008
- *Sierra Katy, Bate Bert*
SCJP Sun Certified Programmer for Java 6 Study Guide.
McGrawHill
- *Dean John, Dean Raymond*
Introducción a la programación con Java.
Primera Edición.
México
McGrawHill, 2009

Yo, Carlos Emiliano Mendoza Hernández, hago mención que esta práctica fue de mi autoría.