Informe de Laboratorio 01

Tema: Java

Nota

Estudiante	Escuela	Asignatura
Sebastian Arley Chirinos Negrón schirinosne@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Estructura de datos y Algoritmos Semestre: I Código: 1702124

Laboratorio	Tema	Duración
01	Java	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - A	Del 29 Mayo 2023	Al 05 Junio 2023

1. Tarea

- Cree una cuenta de usuario en GitHub usando su correo institucional.

opcional por ahora Configure su cuenta de estudiante (https://education.github.com/pack).

- Cree un nuevo proyecto personal y desarrolle el ejercicio resuelto en clase. Haga 3 commits como mínimo y muéstrelos. Commit para "¡Hola mundo!", otro para "Bienvenida al cursoz otro para imprimir su nombre.
- Cree un proyecto grupal para trabajo colaborativo (de 3 a 5 integrantes).
- Cree un archivo por cada tema del manual de java (https://www.w3schools.com/java/default.asp), haga commit e inluyalo en su informe grupal (Dividanse los temas).
- Cree ramas para cada integrante y cada cierto tiempo una las ramas al main. No elimine nada para evidenciar ramas, main y commits.

2. Equipos, materiales y temas utilizados

- Editor Vim
- Java
- Git

- GitHub
- C.m. Construye responsablemente soluciones haciendo uso de estructuras de datos y algoritmos, siguiendo un proceso adecuado para resolver problemas computacionales que se ajustan al uso de los recursos disponibles y a especificaciones concretas.

3. URL de Repositorio Github

- URL del Repositorio GitHub para clonar o recuperar.
- `https://github.com/BastleyNait/EDA-LAB-B-23A.git`
- URL para el laboratorio 04 en el Repositorio GitHub.
- `https://github.com/BastleyNait/EDA-LAB-B-23A/tree/main/lab01`

4. Programa en java: Test de iq

4.1. Clase Main

- Este un programa Java que utiliza la clase Archivos para leer preguntas de un archivo de texto llamado "TestdeIQ.txtz permite al usuario ingresar respuestas para cada pregunta. A continuación, se explica el flujo general del programa:

Listing 1: Main.java

```java
import java.util.Scanner;

public class Main {
    static Scanner entrada = new Scanner(System.in);

    public static void main(String[] args) {
        // Creamos una instancia de la clase Archivos con el objeto testDeIq
        Archivos testDeIq = new Archivos();

        // Leemos el contenido del archivo "TestdeIQ.txt"
        String texto = testDeIq.leerArchivo("TestdeIQ.txt");

        // Dividimos el texto en un arreglo de lneas utilizando el carcter de nueva lnea "\n"
        String[] lineas = texto.split("\n");

        // Iteramos sobre el arreglo de cuestionario
        for (int i = 0; i < 20; i++) {
            // Obtenemos la pregunta correspondiente al ndice actual
            String pregunta = testDeIq.leerPregunta(lineas, i);

            // Mostramos la pregunta en la consola
            System.out.print(pregunta + "\nRespuesta: ");

            // Leemos la respuesta del usuario desde la consola
            String respuesta = entrada.next();

            //  Aqu  puedes realizar alguna operacin con la respuesta, como compararla con una
                respuesta esperada

```

```
            // Por ejemplo:
            // if (respuesta.equals("respuesta_esperada")) {
            //     // Realizar alguna accin si la respuesta es correcta
            // } else {
            //     // Realizar alguna accin si la respuesta es incorrecta
            // }
        }
    }
}
```

- Motrando la Ejecución del codigo:

4.2. Ejercicio 02a

- Antes de ejecutar el código tenemos que tener en cuenta el importar la función draw del archivo interpreter.py y también todas las funciones de chessPicture:

Listing 2: Ejercicio2a.py

```
from interpreter import draw
from chessPictures import *
#se junta caballo blanco con negro se pone encima de caballo negro con blanco juntos
draw(knight.join(knight.negative()).up(knight.negative().join(knight)))
```

- Motrando la Ejecución del codigo:

4.3. Estructura de laboratorio 04

- El contenido que se entrega en este laboratorio es el siguiente:

```
lab04/
+---EjerciciosDocente
|       defs.py
|       esEscalar.py
|       esPalindromo.py
|       esUnitaria.py
|       numeroPares.py
|       operadoresArit.py
|       pythonClass.py
|       strings.py
|       tablaDeMulti.py
|       test_esEscalar.py
|       test_esUnitaria.py
|       tiposDeDatos.py
|
+---Latex
|   |   .gitignore
|   |   Pweb02_lab04_schirinosne.pdf
|   |   Pweb02_lab04_schirinosne.tex
|   |
|   +---img
|   |       Ejercicio2a.png
|   |       Ejercicio2b.png
|   |       Ejercicio2c.png
|   |       Ejercicio2d.png
|   |       Ejercicio2e.png
|   |       Ejercicio2f.png
|   |       Ejercicio2g.png
|   |       logo_abet.png
|   |       logo_episunsa.png
|   |       logo_unsa.jpg
|   |       pseudocodigo_insercion.png
|   |
|   \---src
|           Ejercicio2a.py
|           Ejercicio2b.py
```

```
|          Ejercicio2c.py
|          Ejercicio2d.py
|          Ejercicio2e.py
|          Ejercicio2f.py
|          Ejercicio2g.py
|          Insertion01.java
|          picture.py
|
\---Tarea-del-Ajedrez
        .gitignore
        chessPictures.py
        colors.py
        Ejercicio2a.py
        Ejercicio2b.py
        Ejercicio2c.py
        Ejercicio2d.py
        Ejercicio2e.py
        Ejercicio2f.py
        Ejercicio2g.py
        interpreter.py
        picture.py
        pieces.py
        prueba.py
```

5. Pregunta: ¿Qué son los archivos *.pyc?

- Los archivos .pyc son archivos de código compilado en Python. Cuando un archivo fuente de Python (.py) se ejecuta, el intérprete de Python compila ese código en bytecode, que es una representación intermedia del código que puede ser ejecutada más rápido por la máquina virtual de Python. Los archivos *.pyc contienen este bytecode compilado y se generan automáticamente cuando se importa un módulo en Python.

6. Pregunta: ¿Para qué sirve el directorio pycache?

- El directorio "pycache.[es] un directorio que se crea automáticamente en Python 3 para almacenar los archivos *.pyc. Cuando se importa un módulo en Python, el intérprete buscará si existe un archivo *.pyc correspondiente en el directorio "pycache". Si lo encuentra y es más reciente que el archivo *.py fuente, el intérprete utilizará el archivo *.pyc en su lugar para ahorrar tiempo de compilación. Si no existe un archivo *.pyc o está desactualizado, el intérprete generará uno nuevo.

7. Pregunta: ¿Cuáles son los usos y lo que representa el subguión en Python?

- En cuanto al subguión en Python, se le conoce como underscore y se utiliza de diferentes formas:
- Nombres de variables especiales: En Python, el subguión se utiliza para nombres de variables especiales que tienen un significado específico. Por ejemplo, un subguión simple se utiliza a menudo como un nombre de variable temporal o como un lugar para ignorar valores que no se necesitan.

- Convención para nombres privados: El subguión doble al inicio de un nombre de variable por ejemplo, nombre se utiliza como convención para indicar que un atributo o método es "privado.[en] Python. No hay verdaderos atributos o métodos privados en Python, pero se considera una convención de estilo no acceder directamente a estos atributos o métodos desde fuera de la clase.
- Uso en importaciones: El subguión se utiliza a menudo en las importaciones de módulos en Python.

8. Rúbricas

8.1. Entregable Informe

Tabla 1: Tipo de Informe

Informe	
Latex	El informe está en formato PDF desde Latex, con un formato limpio (buena presentación) y facil de leer.

8.2. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumplio con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos lo items.
- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Tabla 2: Niveles de desempeño

	Nivel			
Puntos	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
2.0	0.5	1.0	1.5	2.0
4.0	1.0	2.0	3.0	4.0

Tabla 3: Rúbrica para contenido del Informe y demostración

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	x	3	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	2	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	2	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
6. Fechas	Las fechas de modificación del código fuente estan dentro de los plazos de fecha de entrega establecidos.	2	X	2	
7. Ortografía	El documento no muestra errores ortográficos.	2	X	2	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	x	4	
Total		20		19	

9. Referencias

- https://www.w3schools.com/python/python_reference.asp
- https://docs.python.org/3/tutorial/