



PRUEBA 2
PROGRAMACIÓN (DAIS-200)
PABLO SALAS

II SEMESTRE 2022 – DURACIÓN: 2 HORAS 45 MINUTOS

Ejercicio 1 (25%)

El centro NutriKi líder en nutrición infantil, anteriormente había solicitado a los estudiantes de Programación ayuda con un programa que les permitiera ingresar y almacenar sus datos. Debido a la gran demanda de citas, es vital que también el programa genere un reporte con estadísticas e información relevante para la toma de futuras decisiones.

Para llevar a cabo este desafío, NutriKi facilita el programa `nutriki.py` que tiene implementado un menú con 3 opciones:

1. Ingresar datos

El programa solicita el nombre del paciente, nombre del nutricionista, edad del paciente y el horario (mañana-tarde). Ahora en el nuevo programa también debe solicitar el tipo de paciente (nuevo-antiguo) y el estado de la cita (confirmada-suspendida).

2. Estadísticas

Ya está implementado:

- Cantidad total de pacientes ingresados

Falta por implementar:

- Porcentaje de pacientes que suspenden su cita
- Promedio de edad de los pacientes
- Horario con más citas
- Desplegar lista de pacientes nuevos

3. Salir

El programa termina sólo si el usuario ingresa a esta opción. Esta opción ya está implementada.

Nota:

En `nutriki.py` hay comentarios donde usted debe incluir el código que implemente lo solicitado.

Debe utilizar listas para almacenar los datos, validar las entradas y manejar las excepciones para que no se produzcan errores durante la ejecución.



Ejercicio 2 (35%)

Implemente un programa con funciones que lea la siguiente información relacionada con planetas desde un archivo:

	method	number	orbital_period	mass	distance	year
0	Radial Velocity	1	269.300000	7.10	77.40	2006
1	Radial Velocity	1	874.774000	2.21	56.95	2008
2	Radial Velocity	1	763.000000	2.60	19.84	2011
3	Radial Velocity	1	326.030000	19.40	110.62	2007
4	Radial Velocity	1	516.220000	10.50	119.47	2009
...
1030	Transit	1	3.941507	NaN	172.00	2006
1031	Transit	1	2.615864	NaN	148.00	2007
1032	Transit	1	3.191524	NaN	174.00	2007
1033	Transit	1	4.125083	NaN	293.00	2008
1034	Transit	1	4.187757	NaN	260.00	2008

1035 rows x 6 columns

Donde:

- method: método de recopilación de datos
- number: número de avistamientos
- orbital_period: periodo orbital, es tiempo que le toma a un astro recorrer su órbita
- mass: masa del objeto visto
- distance: distancia del astro
- year: año de avistamiento

Entregue las siguientes estadísticas usando funciones:

- Promedio del periodo orbital y de masa por método y promedio del periodo orbital por años
- Cantidad total por métodos

Nota: implemente la lectura de datos y asignación de datos, además de las estructuras necesarias para responder las estadísticas, no olvidar de validar el ingreso de información.

Ejercicio 3 (40%)

En la facultad de ingeniería existen distintos tipos de ayudantes los cuales desempeñan labores que tienen cierta cantidad de horas. Se necesita gestionar el pago que se le asignan a los ayudantes además de obtener cierta información necesaria para futuras contrataciones.

Es por esto que se han recopilado distintos datos de los ayudantes en 2 archivos de texto. El primero se llama Ayudantes.txt que contiene los datos individuales de cada ayudante y el segundo Ayudantias.txt el cual tiene los datos asociados a las ayudantías de los tutores.



El formato de cada uno es el siguiente:

Ayudantes.txt

```
Felipe Salas,A,23  
Javiera Avello,C,20  
Ignacia Ortego,B,21  
Oscar Contreras,A,19
```

Formato: nombre,tipo,edad

Ayudantias.txt

```
Oscar Contreras,6,Electrotecnia  
Javiera Avello,10,Programacion  
Oscar Contreras,10,Programacion  
Felipe Salas,15,Arquitectura de
```

nombre,cantidad_horas,nombre_ramo

Antes de obtener las estadísticas es necesario gestionar el pago que se le asigna a cada ayudante. Para esto se debe calcular la ganancia por cada ayudantía de la siguiente forma:

Ganancia = cantidad_horas*precios_tipo

Cada precio esta relacionado al tipo de ayudante considerando las siguientes listas (No es necesario usarlas, solo es una referencia):

tipos = [A,B,C]

precios_tipo = [6000,5500,5000]

Si un ayudante tiene mas de 1 ayudantía, se debe calcular la ganancia en cada una y sumarla al total para relacionar esa suma al nombre del ayudante (Se recomienda generar una lista de ganancia).

Finalmente se pide obtener la siguiente información:

1. Nombrar a todos los ayudantes de un ramo especifico (Pedir el nombre del ramo desde pantalla)
2. Porcentaje de ayudantes tipo C respecto al total.
3. Nombre y edad del ayudante con más edad.
4. Desplegar el nombre del ayudante con menor ganancia.

Consideraciones:

- Se debe crear un subprograma para leer los archivos y otro para generar una lista con las ganancias totales de cada ayudante.
- Los requerimientos deben ser resueltos utilizando listas.