***EJERCICIO FUNCIONES Nº 4***

Complete el programa presentado en la Figura 1, construyendo las funciones llamadas o invocadas. Considere que el programa simula la postulación a las carreras de una universidad, indicándole al postulante aquellas carreras a las que puede postular de acuerdo a su puntaje ponderado y al último matriculado en la carrera durante el año anterior. El programa debe simular la postulación en todas las carreras y determinar a cuales puede postular. En la figura 2, se muestra la información del postulante que se debe considerar para calcular el puntaje ponderado. Para simplificar el problema, el programa se ejecutará en una universidad con 10 carreras. Debe considerar que para cada carrera, los porcentajes de cada puntaje son distintos, por lo cual deben ser ingresados, al inicio, por el usuario, de acuerdo al formato presentado en la figura 3.

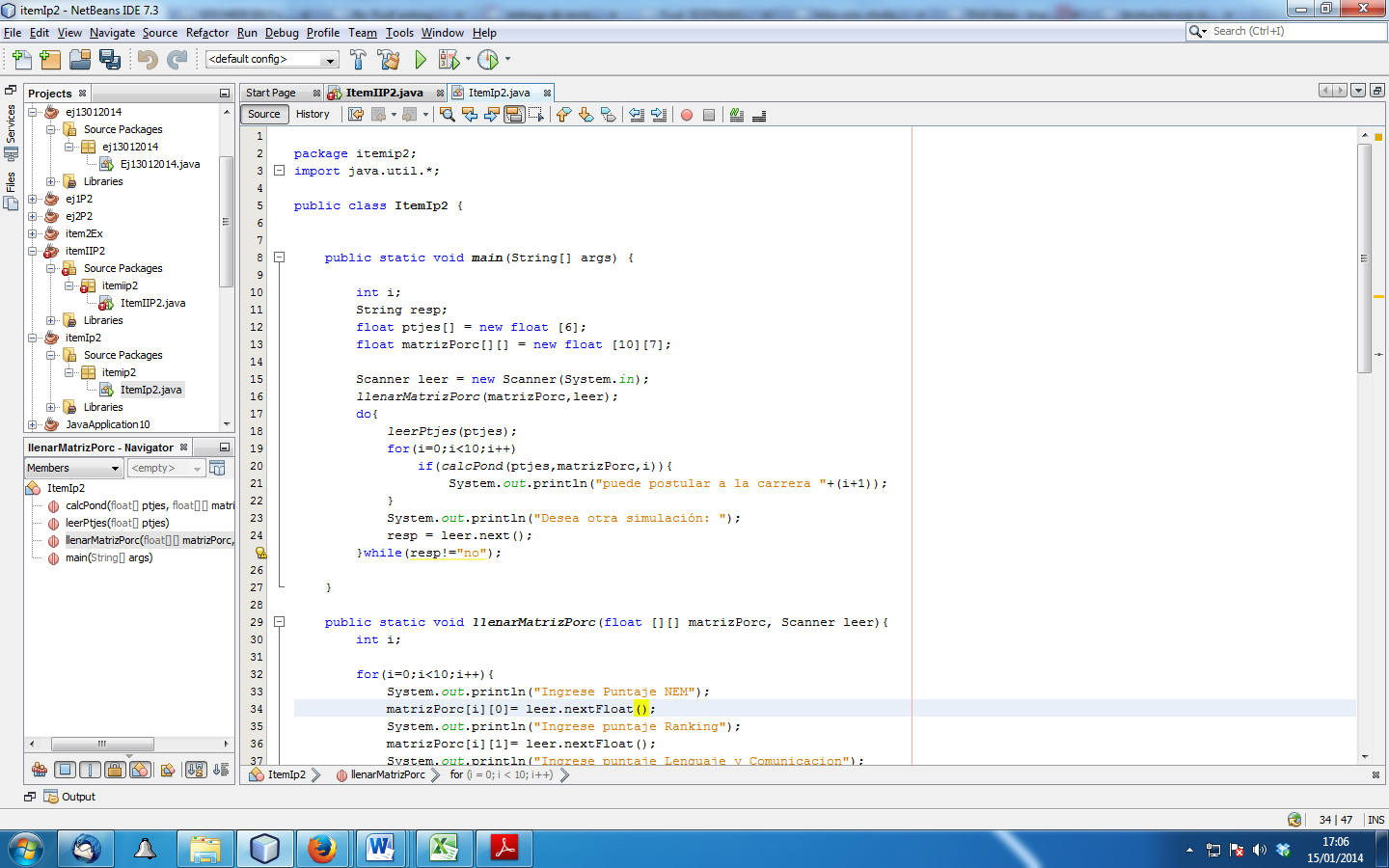
***Figura 1: Código de la función principal del programa.***

Considere además que:

1. El puntaje ponderado se calcula de la siguiente manera:

Puntaje Ponderado = Σ (Puntaje\_Item\* Porcentaje\_Item)

1. Se puede postular a una carrera si el puntaje ponderado es mayor o igual al puntaje del último matriculado del año anterior.



***Figura 2: datos del usuario que deben ser recogidos para realizar la simulación.***

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Puntaje NEM | Puntaje Ranking | Puntaje Prueba Lenguaje y Comunicación | Puntaje Prueba Matemáticas | Puntaje Prueba Historia y Ciencias sociales | Puntaje Prueba Ciencias |

***Figura 3: Matriz de porcentajes, considerados en la simulación.***

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Porcentaje NEM | Porcentaje Ranking | Porcentaje Prueba Lenguaje y Comunicación | Porcentaje Prueba Matemáticas | Porcentaje Prueba Historia y Ciencias sociales | Porcentaje Prueba Ciencias | Puntaje Último Matriculado |

* *llenarMatrizPorc*: Función que ingresa los porcentajes de cada item mostrado en la figura 3, por cada carrera de la universidad. **(15 Pts.)**
* *leerPtjes*: Función que lee los puntajes de cada postulante, según los ítemes mostrados en la figura 2. **(15 Pts.)**
* *calcPond*: Función que calcula el puntaje ponderado de cada postulante y determina si puede o no postular a la carrera. **(30 Pts.)**

*Importante:*

1. *Debe ajustarse obligatoriamente a la estructura de programa presentada en la figura, es decir, la función principal no puede ser cambiada.*
2. *Debe validar los datos de entrada*
3. *Asuma el enunciado como cierto, no realice mejoras al programa principal ni modificaciones en las especificaciones de las funciones.*

***RESPUESTA***

public static void llenarMatrizPorc(float [][] matrizPorc, Scanner leer){ **----- Cabecera, 5 Pts.**

int i,j;

float suma; **----- Validaciones, 3 Pts.**

for(i=0;i<10;i++){ **----- Ciclo, 2 Pts.**

do{

do{

System.out.println("Ingrese Porcentaje NEM");

matrizPorc[i][0]= leer.nextFloat();

} while(matrizPorc[i][0]<0 || matrizPorc[i][0]>100);

do{

System.out.println("Ingrese Porcentaje Ranking");

matrizPorc[i][1]= leer.nextFloat();

} while(matrizPorc[i][1]<0 || matrizPorc[i][1]>100);

do{

System.out.println("Ingrese Porcentaje Lenguaje y Comunicacion");

**Llenado matriz, 5 Pts.**

* **Tipo de datos correcto**
* **Posiciones correctas**
* **Sintaxis correcta**

matrizPorc[i][2]= leer.nextFloat();

} while(matrizPorc[i][2]<0 || matrizPorc[i][2]>100);

do{

System.out.println("Ingrese Porcentaje Matemáticas");

matrizPorc[i][3]= leer.nextFloat();

} while(matrizPorc[i][3]<0 || matrizPorc[i][3]>100);

do{

System.out.println("Ingrese Porcentaje Historia y Ciencias Sociales");

matrizPorc[i][4]= leer.nextFloat();

} while(matrizPorc[i][4]<0 || matrizPorc[i][4]>100);

do{

System.out.println("Ingrese Porcentaje Ciencias");

matrizPorc[i][5]= leer.nextFloat();

} while(matrizPorc[i][5]<0 || matrizPorc[i][5]>100);

suma=0;

for(j=0;j<6;j++)

suma += matrizPorc[i][j];

}while(suma!=100);

do{

System.out.println("Ingrese puntaje último matriclado agno anterior");

matrizPorc[i][6]= leer.nextFloat();

} while(matrizPorc[i][6]<0 || matrizPorc[i][6]>850);

}

}

public static void leerPtjes(float [] ptjes){ **----- Cabecera, 5 Pts.**

float suma; **----- Validaciones, 3 Pts.**

Scanner leer2 = new Scanner(System.in); **----- Definición objeto Scanner, 2 Pts.**

do{

System.out.println("Ingrese Puntaje NEM");

ptjes[0]= leer2.nextFloat();

} while(ptjes[0]<0 || ptjes[0]>850);

do{

System.out.println("Ingrese puntaje Ranking");

ptjes[1]= leer2.nextFloat();

} while(ptjes[1]<0 || ptjes[1]>850);

do{

**Llenado matriz, 5 Pts.**

* **Tipo de datos correcto**
* **Posiciones correctas**
* **Sintaxis correcta**

System.out.println("Ingrese puntaje Lenguaje y Comunicacion");

ptjes[2]= leer2.nextFloat();

} while(ptjes[2]<0 || ptjes[2]>850);

do{

System.out.println("Ingrese puntaje Matemáticas");

ptjes[3]= leer2.nextFloat();

} while(ptjes[3]<0 || ptjes[3]>850);

do{

System.out.println("Ingrese puntaje Historia y Ciencias Sociales");

ptjes[4]= leer2.nextFloat();

} while(ptjes[4]<0 || ptjes[4]>850);

do{

System.out.println("Ingrese puntaje Ciencias");

ptjes[5]= leer2.nextFloat();

} while(ptjes[5]<0 || ptjes[5]>850);

}

public static boolean calcPond(float [] ptjes, float [][] matrizPorc, int valor){ **----- Cabecera 10 Pts.**

float ptjePost=0;

int i,j;

for(j=0;j<6;j++){ **----- Cálculo puntaje Ponderado 10 Pts.**

ptjePost += ptjes[j]\*matrizPorc[valor][j];

}

if (ptjePost>=matrizPorc[valor][6]) **----- Comparación 5 Pts.**

return true; **----- Retornos 5 Pts.**

else

return false;

}