

UNIVERSIDAD DE ANTIOQUIA
PROGRAMA INGENIERÍA DE SISTEMAS
TÉCNICA DE PROGRAMACIÓN Y LABORATORIO G1.
PARCIAL
PROF: ING GEOVANNY MENDOZA GONZALEZ

1. Objetivo

El objetivo de este parcial es comprobar su familiaridad con el uso de las clases, la herencia, la interfaz y la E/S de archivos.

El archivo debe ser enviado por la plataforma: "Escuela.java"

2. Requisitos

- a) El parcial debe realizarse con los métodos y clases especificados (no, sólo con un único main()) Tenga en cuenta que cada programa requiere la escritura de métodos y clases, y una rutina principal para implementar esas clases y métodos.
- b) NO hay entrada de teclado/salida de pantalla especificada en los métodos mismos - esto significa que NO debe tener ninguna declaración de impresión o uso de Scanner dentro de estos métodos nombrados. Cualquier impresión y/o entrada de teclado es realizada por main() en cada ejercicio
- c) Por favor, añada su nombre y apellido como comentarios en la parte superior de sus programas.
- d) Por favor, añada comentarios adecuados en el programa para explicar sus ideas.
- e) Por favor, haga coincidir el patrón de salida de las ejecuciones de muestra tanto como sea posible.

3. Escuela.java

El objetivo principal de este proyecto es construir una estructura para almacenar la información de los estudiantes. Es necesario escribir varias interfaces/clases, y declarar un array en main() para almacenar la información. Se ofrece un archivo input.txt que se adjunta en el parcial, este archivo contiene toda la información. Necesitarás leer desde ese archivo a tu array en main(), hacer algunos cálculos, y sacar el resultado en un archivo output.txt.

- a) Construye una clase **abstracta** llamada "**Estudiante**". Declara una variable llamada **nombre** de tipo String, un variable tipo char llamada **gradoLetra**. Construye aquí los getter (getNombre(), getGradoLetra()) y declara aquí el setter setGradoLetra() como método abstracto.
- b) Construya una subclase derivada de "Estudiante", llamada "**Undergrade**". Incluye un atributo más, llamado **gpa** y de tipo double.
Escribe un constructor **Undergrade**(String nombre, double a, double b, double c). El String es para inicializar el **nombre** del estudiante de "Estudiante". Y **gpa** se inicializa como la puntuación media de estos tres dobles de entrada.
Realice el **setGradoLetra()** como para establecer el **GradoLetra** basado en el gpa actual.

GPA	GradoLetra
[90, 100]	a
[80, 90]	b
[70, 80]	c
[60, 70]	d
De lo contrario	f

- c) Construye una subclase derivada de "**Estudiante**", llamada "**Graduado**". Incluye tres booleanos más: **pass1**, **pass2**, **pass3**.

La calificación de un estudiante graduado se decide por el trabajo que ha presentado. Tres comités darán respuesta al trabajo como 'Aprobado' o 'No'. Escriba un constructor Graduado(String nombre, char a, char b, char c). El String es para inicializar el **nombre** del estudiante de "**Estudiante**". **pass1**, **pass2**, **pass3** son inicializados por los tres caracteres: si uno es igual a '**P**' significa que un comité de '**Aprobado**', se inicializa uno de **pass1**, **pass2**, **pass3** como Verdadero. Por ejemplo, si a es 'P', b es 'P', c es 'N', puede inicializar pass1 como verdadero, pass2 como verdadero, pass3 como falso.

Los caracteres de entrada pueden ser cualquier letra, todo lo que no sea 'P' debe ser tomado como no aprobado.

- d) Realice el método **setGradoLetra()** estableciendo **GradoLetra** como 'P' sólo cuando todos los comités de 'Aprobado' (todos verdaderos para sus atributos de Aprobado). De lo contrario, se establece como 'N'.
- e) En main(): construye un array de **Estudiantes** de tamaño 10. Tenemos 5 estudiantes no graduados(**Undergrade**) y 5 de graduado. Su información se ofrece en "input.txt" en el archivo adjunto.

Descárgalo. Las primeras 5 líneas son detalles para los estudiantes no graduados "Undergrade" y las siguientes 5 líneas son detalles para los graduados. Inicialice su array con la información ofrecida en input.txt. No debe escribirlos en su main(), sino usar el FileInputStream o FileReader para obtener la información del archivo. Compruebe primero el formato de la información en input.txt, y luego gestione sus bucles for para leer de él correctamente.

Utiliza un bucle para calcular el **GradoLetra** de cada estudiante, e imprime su nombre y el **GradoLetra** en un nuevo archivo llamado output.txt, a través de FileOutputStream o FileWriter. El formato debe ser:

```
NombreAlumno1 letraCalificación1
NombreAlumno2 letraCalificación 2
```

- f) Al enviar, sólo envíe el archivo .java. No envíe el output.txt.

Ejemplo input.txt:

```
Ketty 88 95 73
Valeria 73 66 85
Jhonatan 93 95 86
Jhon 54 62 38
Karen 79 83 91
Ana P P P
Mauricio N P N
Cristian P A P
Esteban P P P
Liz A N P
```

Salida relacionada.txt:

```
Ketty b
Valeria c
Jhonatan a
Jhon f
Karen b
Ana P
Mauricio N
Cristian N
Esteban P
Liz N
```

4. Directrices genéricas de calificación

- Por favor, asegúrese de que sólo está utilizando los conceptos ya discutidos en clase. Es decir, intenta limitarte a bucles, sentencias de selección, métodos String que hablen de la clase, arrays y tus propios métodos escritos.
- No es necesario que realices ningún tipo de comprobación del tipo de entrada.
- Por favor, asegúrese de que se ajusta a las especificaciones (nombre del programa, sentencias de impresión, entradas y salidas esperadas, etc.).
- Por favor, asegúrese de que su código es legible. Añade comentarios adecuados a tu código.
- Por favor, asegúrese de que ha compilado y ejecutado su programa antes de entregarlo. Se le otorgará una calificación de 0 si su programa no compila.