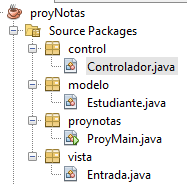
**Multi302 Febrero 14 Aplicando MVC con POO**

**Temas Tratados**

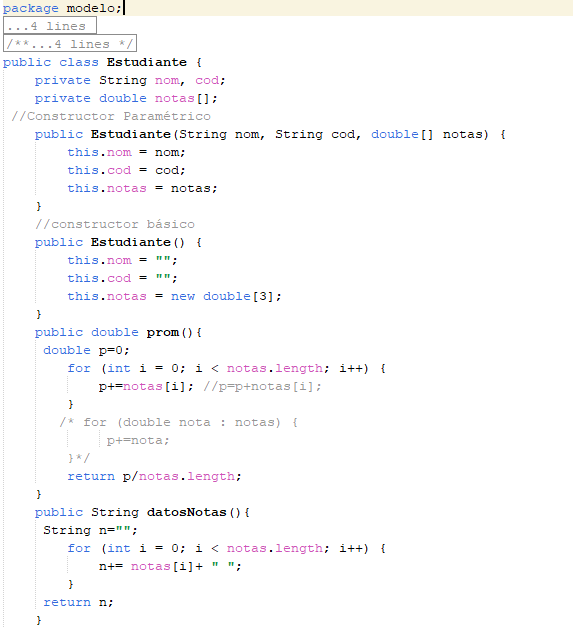
1. Continuación ejercicio ProyNotas
2. Aplicando MVC a ejercicio ProySeguro

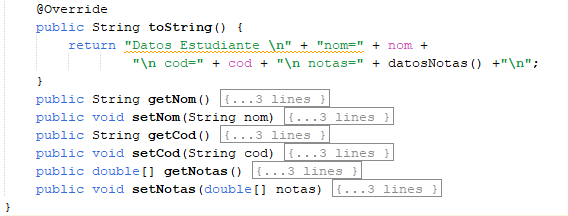
**Desarrollo**

1. Continuación ejercicio ProyNotas

En este ejercicio se está aplicando el concepto de MVC, utilizando la siguiente estructura:

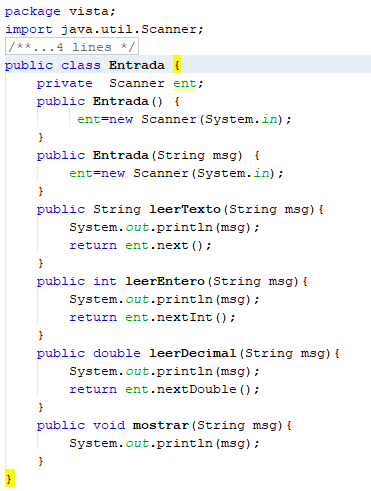
**Paquete Modelo,** en este paquete se crearan las clases que contienen la estructura y la lógica del negocio.  
**Clase Estudiante**





**Paquete Vista**: en este paquete se agregarán las clases que permiten realizar los procesos de interacción (Mostrar o capturar los datos) ,para ello se crea las clase Entrada.

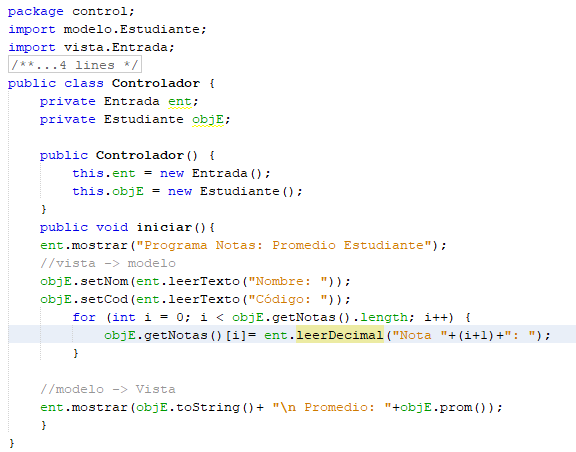
**Clase Entrada**



Observe que al definir el objeto **Scanner**, los métodos **leerTexto(), leerEntero() y leerDecimal()** recibirán un mensaje que contendrá el texto que observará el usuario indicando que dato deberá digitar y mediante los métodos **next(), nextInt() y nextDouble()** se podrán obtener los datos digitados, los cuales serán retornados para que sean utilizados por el controlador, ya sea para presentarlos en pantalla o pasarlos al modelo.

**Paquete Control:** En este paquete se definirá la clase Controlador, que se encargará de manipular el flujo de información entre la capa de la vista (Clase Entrada) y el modelo (Clase Estudiante)

**Clase Controlador**

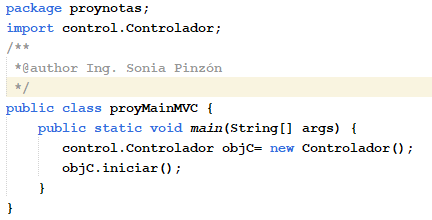


Observe que el controlador en el metodo iniciar utiliza el objeto ent para usar los métodos leerTexto() y leerDecimal para enviar la información al objeto objE. En este caso se está pasando de la vista al modelo

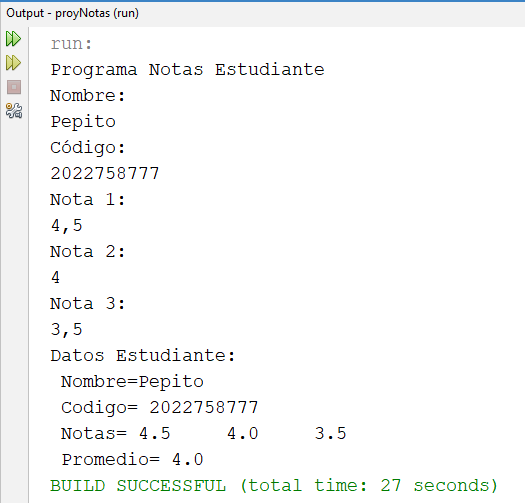
Y para presentar la información se está extrayendo la información del objeto Estudiante objE mediante los métodos .toString() y .prom() para mostrarlos en pantalla, de esta forma el controlador pasa del modelo a la vista.

**Clase ProyMainMVC**

Finalmente, para ejecutar el programa, se crea la clase autónoma que contiene el método main. En esta se está Agregando un objeto de tipo Controlador para ejecutar el método iniciar().



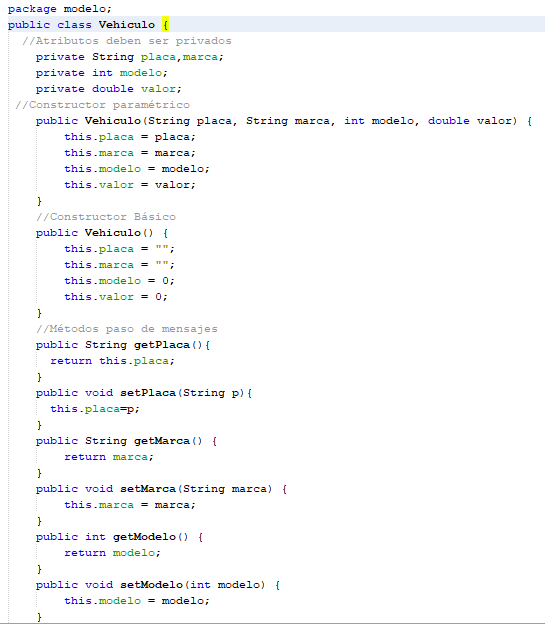
Salida de ejecución Shift+F6

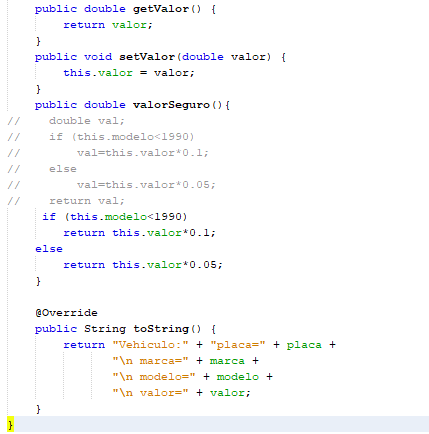


1. **Ejercicio ProySeguro en MVC**

El objeto de este ejercicio es calcular el valor a pagar del impuesto de ún vehiculo teniendo encuenta que los vehiculos modelos superiores a 1990 pagan el 5% del valor del vehiculo y los demás pagarán el 10%. (PAra este ejercicio se puede copiar y agregar la clase Entrada del proyecto anterior)

**Clase Vehículo (Modelo)**

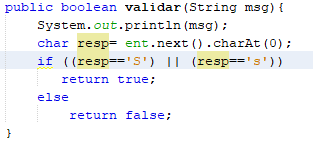




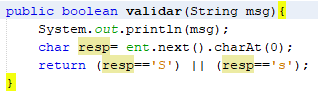
\*Los métodos get y set se generan automáticamente (Alt+Insert)

**Clase Controlador (Control)**

En esta Clase se está agregando un Ciclo con centinela, para validar si el usuario desea cagregar otro vehiculo. La validación se realiza mediante el método validar() que se deberá agregar en la clase Entrada con el siguiente código.

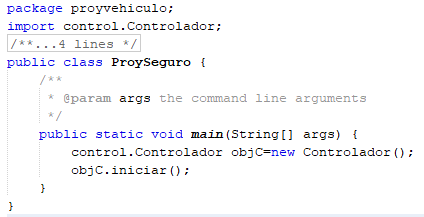


También se puede simplificar el código de la siguiente manera





**Clase Principal ProySeguro**



**Próxima Clase**

- Revisar material ( aula virtual Moodle).

- Implementar proyecto Ejercicio Empleado. aplicando el MVC.

- Revisar conceptos Relaciones entre clases Composición y Agregación. Practicar, Practicar y practicar...