

<b>Nombre Asignatura:</b>	Programación Avanzada	<b>Prueba:</b>	1 - Práctica
<b>Nombre Estudiante:</b>		<b>Rut:</b>	
<b>Nombre Docente:</b>	Tulio Enrique León Alcalá		
<b>Nivel:</b>	III	<b>Fecha:</b>	26-04-2021
<b>Puntaje Obtenido:</b>		<b>Nota:</b>	
<b>REVISIÓN DE LA PRUEBA</b>			
<b>Fecha de revisión:</b>		<b>Firma del Estudiante:</b>	

**Resultados de Aprendizaje a evaluar en este instrumento:**

Resuelve problemas de programación de complejidad básica, aplicando la Programación Orientada a Objetos, a través del análisis crítico para generar posibles alternativas de solución.

**Criterios de evaluación de este instrumento:**

**CE 1:** Emplea conceptos de programación orientada a objetos en la resolución de problemas básicos generando una posible solución.

**CE 2:** Propone soluciones a través al análisis crítico y el lenguaje Java, utilizando tipos abstractos de datos.

**CE 3:** Construye soluciones informáticas en lenguaje Java utilizando estructuras de datos estáticas, tipos abstractos de datos y el paradigma orientado a objetos.

**Descripción del instrumento:**

Este instrumento se compone de un problema planteado (60%) de la nota Prueba Solemne 1.

## PROGRAMACIÓN I - PRUEBA SOLEMNE Nro. 1 – 2da. Parte (Práctica)

---

La prueba solemne parte práctica de Programación Avanzada, es de carácter personal, corresponde a un proyecto a desarrollar en lenguaje Java dando solución a un problema propuesto por el estudiante.

### INSTRUCCIONES: (Revise con detenimiento)

- El formato de entrega es .zip (proyecto exportado en NetBeans). El nombre del Proyecto como del archivo exportado debe ser con el siguiente formato: “Apellido\_ Nombre\_P1” (Donde Apellido y Nombre debe ser sustituido por el Apellido y Nombre del integrante). *Las instrucciones para exportar un Proyecto se encuentran en el Aula Virtual.*
- Suba el archivo exportado junto a una captura de pantalla del resultado de la ejecución a la plataforma donde se encuentra esta prueba. **Sólo en caso extremo** de problemas envíelo a tulio.leon@cloud.uautonoma.cl
- Para cualquier consulta sobre la prueba la puede hacer en el Foro para preguntas y respuestas, pero las preguntas deben ser de forma general, no específica para cada proyecto, aclarando los puntos mencionados más adelante.

**NOTA: LA PRUEBA ES DE CARÁCTER INDIVIDUAL, EN CASO DE COPIA CON OTRO ALUMNO, YA SEA DE ESTA SECCIÓN O DE OTRA, SERÁ CALIFICADO CON 1. Tampoco se permite tomar ejemplos de los colocados en el material de clases o resueltos como ejercicios.**

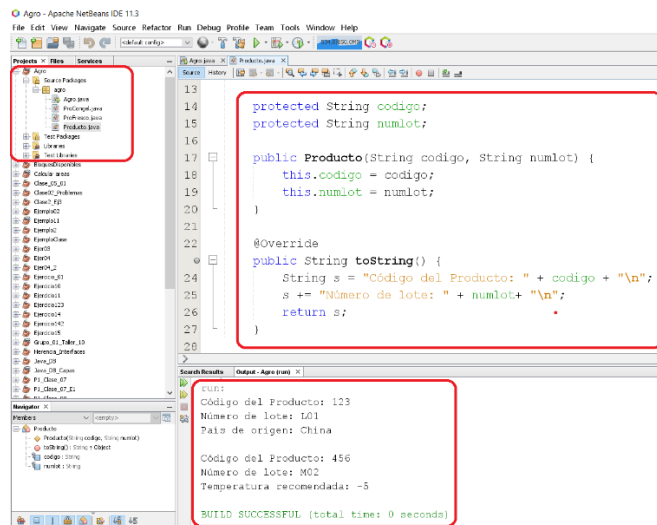
### PLANTEAMIENTO (Leer con detenimiento)

El problema o tema para abordar en el proyecto lo decide el alumno, (no olvides presentar tu planteamiento del problema en la clase principal como las primeras líneas en calidad de comentario. Se busca que demuestres que aplicas los conocimientos de programación orientada a objetos.

Dicho proyecto debe considerar: Tú debes decidir el tema a tratar para implementar todo lo solicitado.

1. Creación de una clase para crear los objetos pertinentes del problema planteado.
2. La clase debe tener al menos 4 atributos con diversos tipos: String, int, char, float, etc. y no debe exceder de 6 atributos.
3. La clase debe tener todos sus métodos constructores (sin parámetros y con todos los parámetros), y los métodos getters y setters.
4. La clase debe tener un método (no void) que **realice cálculos según una condición (usar if) de uno o más atributos** y retorne (return) el tipo de valor del método.
5. La clase debe tener un método (void) que muestre por pantalla todos los valores de los atributos, y adicionalmente mostrar el valor calculado del punto 4.
6. En el método principal main() (que debe estar dentro de la clase principal que tiene el nombre del proyecto) se deben leer (usando Scanner) los datos de entrada para 3 entidades (usar ciclo) siguiendo igual en los puntos 7 a 8. (Recomendación: Hágalo para una sola entidad y luego que funcione implemente el ciclo para las 3 entidades)

7. En el método principal main() se debe crear un objeto una vez leído los datos para cada entidad, es decir, con el ciclo se deben leer los datos para 3 entidades pero cada vez que se hayan leído los datos para una entidad, se debe proceder a crear el respectivo objeto con esos datos.
8. En el método principal main() cada vez que sea creado un objeto se debe llamar al método que muestra los datos (punto 5). Debe tomar una captura de pantalla (Print Screen) con el resultado de al menos 1 entidad junto con parte del código de su Proyecto en NetBeans. Ejemplo:



9. Su Proyecto no debe presentar errores de compilación, se debe ejecutar correctamente y debe mostrar los resultados correctos de acuerdo con los datos que introduzca el usuario.
10. La entrega del proyecto debe ser en formato de archivo exportado de NetBeans (.zip), de no ser así corre el riesgo de que su Proyecto sea evaluado con nota 1.

**Nota:** El docente tendrá derecho, en caso de ser necesario, a exigir una defensa del Proyecto por parte del estudiante.

## RÚBRICA PARA EVALUAR SOLEMNE 1 – 2da PARTE

Dimensiones	¿Qué se pide?	Pond	Destacado 7 puntos	Habilitado 5 puntos	En desarrollo 3 puntos	Insuficiente 1 punto
Resuelve problemas de programación de complejidad básica, aplicando la Programación Orientada a Objetos, a través del análisis crítico para generar posibles alternativas de solución.	(1) Propone soluciones a través al análisis crítico y el lenguaje Java, utilizando tipos abstractos de datos.	10%	El diseño y análisis de las clases que propone el estudiante: a) Define atributos atingentes b) Define métodos	El diseño y análisis de las clases que propone el estudiante: a) Define atributos atingentes b) Define métodos con funcionalidades que no utilizan los atributos de la clase	El diseño y análisis de las clases que propone el estudiante es insuficiente: a) Define atributos que no representa a las propiedades de la clase b) Define métodos con funcionalidades que no utilizan los atributos de la clase	El diseño y análisis de las clases que propone el estudiante no es satisfactorio: a) Define atributos que no representa a las propiedades de la clase o no cumple con el concepto de herencia. b) No define métodos con funcionalidades que realicen alguna operación
	(2) La clase debe tener al menos 4 atributos con diversos tipos: String, int, char, float, etc. y no debe exceder de 6 atributos.	5%	La Clase tiene mínimo 4 atributos de diferentes tipos y no excede de 6 atributos.	La Clase tiene mínimo 4 atributos, pero solo varían en 3 tipos diferentes y no excede de 6 atributos.	La Clase tiene mínimo 4 atributos, pero solo varían en 2 tipos diferentes o solo tiene 3 atributos.	La Clase tiene mínimo 4 atributos, pero todos son del mismo tipo, o tiene menos de 3 atributos o excede los 6 atributos.
	(3) La clase debe tener todos sus métodos constructores (sin parámetros y con todos los parámetros), y los métodos getters y setters.	5%	La clase tiene todos sus métodos constructores (sin parámetros y con todos los parámetros), y los métodos getters y setters.	La clase tiene todos sus métodos constructores, pero no tiene los métodos getters ni setters.	La clase no tiene todos los métodos constructores, pero tiene los métodos getters y setters.	La clase no tiene todos los métodos constructores ni los métodos getters y setters.
	(4) La clase debe tener un método (no void) que realice cálculos según una condición (usar if) de uno o más atributos y retorne (return) el tipo de valor del método.	15%	La clase tiene un método (no void) que realiza cálculos según una condición de uno o más atributos y retorna el tipo de valor del método.	La clase tiene un método (no void) que realiza cálculos pero no sobre una condición de uno o más atributos y retorna el tipo de valor del método.	La clase tiene un método (no void) que realiza cálculos pero no sobre una condición de uno o más atributos pero no retorna un valor correcto.	La clase no tiene un método (no void) que retorne un valor.
	(5) La clase debe tener un método (void) que muestre por pantalla todos los valores de los atributos, y adicionalmente mostrar el valor calculado del punto 4.	15%	La clase tiene un método (void) que muestra por pantalla todos los valores de los atributos, y adicionalmente mostrar el valor calculado del punto 4.	La clase tiene un método (void) que muestra por pantalla todos los valores de los atributos, pero no muestra el valor calculado del punto 4.	La clase tiene un método (void) que no muestra por pantalla todos los valores de los atributos, pero si muestra el valor calculado del punto 4.	La clase no tiene un método (void) que muestre por pantalla los valores de los atributos ni el valor calculado del punto 4.
	(6) En el método principal main() (que debe estar dentro de la clase principal que tiene el nombre del proyecto) se deben leer los datos de entrada para 3 entidades (ciclo)	15%	Contiene una estructura para leer todos los datos de entrada para 3 entidades	Lee todos los datos de entrada para una sola entidad	Lee solo algunos datos de entrada para una sola entidad	No lee ningún dato de entrada para ninguna entidad
	(7) En el método principal main() se debe crear un objeto una vez leído los datos para cada entidad.	10%	Crea un objeto de forma correcta para cada entidad con todos sus valores	Crea un objeto de forma correcta para solo una entidad con todos sus valores	Crea los objetos de forma incorrecta para una o más entidades	No crea ningún objeto

	(8) En el método principal main() una vez creado el objeto se debe llamar al método que muestra los datos (punto 5).	10%	Llama al método de cada objeto de forma correcta.	Llama al método para un solo objeto de forma correcta.	Llama al método para todos los objetos de forma incorrecta.	No llama a ningún método.
	(9) Compilación, Ejecución y resultados esperados del proyecto	10%	Presenta cero errores al compilar el proyecto y los resultados son correctos, los cuales se verifican en la captura de pantalla.	Presenta cero errores al compilar el proyecto y los resultados son incorrectos, los cuales se verifican en la captura de pantalla.	Presenta pocos errores al compilar el proyecto y los resultados son incorrectos o no adjunta la captura de pantalla.	Presenta muchos errores de compilación
	(10) La entrega del proyecto debe ser en formato de archivo exportado de NetBeans (.zip)	5%	Entrega el proyecto exportado con el nombre correcto del Proyecto igual al archivo exportado.	Entrega el proyecto exportado con el nombre correcto del Proyecto, pero con nombre incorrecto del archivo exportado.	Entrega el proyecto exportado con el nombre incorrecto del Proyecto, pero con nombre correcto del archivo exportado.	No exporta bien el proyecto