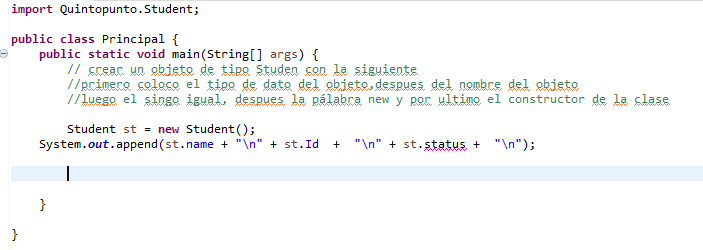
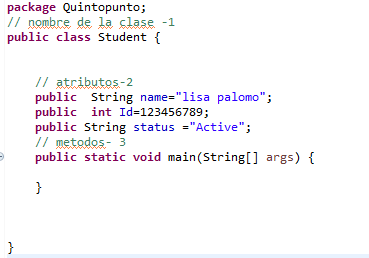
**Taller 1 Nombre: Nicolás Arteaga Arteaga código: 1114541427**

**1-** 

**2- ¿Describa la diferencia entre el formato CamelCase en mayúsculas y en minúsculas y proporcione un ejemplo de cuándo los usaría??**

CamelCase es un estilo de escritura que se aplica a frases o

palabras compuestas. El nombre se debe a que las mayúsculas a lo

largo de una palabra en CamelCase se asemejan a las jorobas de

un camello. El nombre CamelCase se podría traducir como

Mayúsculas/Minúsculas Camello. El término case se traduce como

"caja tipográfica", que a su vez implica si una letra es mayúscula o

minúscula y tiene su origen en la disposición de los tipos móviles en casilleros o cajas.

Existen dos tipos de CamelCase:

• UpperCamelCase, cuando la primera letra de cada una de las palabras es mayúscula. Ejemplo:

EjemploDeUpperCamelCase.

• lowerCamelCase, igual que la anterior con la excepción de que la primera letra es minúscula. Ejemplo:

ejemploDeLowerCamelCase.

3**¿Qué sintaxis se utiliza para importar todo el paquete de utilidades de Java? Y si importase todo el paquete, ¿también necesitaría importar las clases adicionales en el mismo paquete por separado?**

La funcionalidad de una aplicación Java se implementa habitualmente en múltiples clases, entre las que suelen existir distintas relaciones. **Las clases se agrupan en unidades de un nivel superior, los paquetes**, que actúan como **ámbitos de contención de tipos**. Cada módulo de código establece, mediante la palabra clave package al inicio, a qué paquete pertenece. Como ya sabrás, con la **cláusula import** cualquier módulo de código puede hacer referencia a tipos definidos en otros paquetes.

Vamos a familiarizarnos con los procedimientos para construir paquetes, para referenciarlos desde otros puntos y también veremos la **relación existente entre paquetes, clases y el sistema de archivos** donde se almacena el proyecto.

6- **\* public**

* **public** **class**
* **int**
* **package**
* **static**
* **void**
* **new**
* **impor**
* **boolean**
* **return** **super**