

Antes de comenzar me gustaría aclarar que no se porqué en eclipse se me borran paquetes solos o me aparecen signos ">" como "?" de la nada en muchas clases. Todo se encuentra implementado en el repo <https://github.com/ManuelMichellon/unqui-po2-michellon>, inclusive este documento.

1. Contador de pares, impares y múltiplos

Cree un nuevo proyecto de Java en Eclipse. Cree una clase llamada `Counter` dentro del paquete `unq` que, a partir de un `ArrayList` que mantiene como variable privada, sea capaz de contar:

1. La cantidad de pares
2. La cantidad de impares
3. La cantidad de múltiplos de un cierto número.

RTA: Repo -> <https://github.com/ManuelMichellon/unqui-po2-michellon>

2. Examinar las expresiones

Dado el siguiente código:

```
String a = "abc";

String s = a;

String t;
```

Indique que valores retornan las siguientes expresiones o, si dan error, por qué se producen:

- `s.length();` -> **3**
- `t.length();` -> **ERROR: T es un string pero no está instanciada para pedirle lenght()**
- `1 + a;` -> **1abc**
- `a.toUpperCase();` -> **ABC**
- `"Libertad".indexOf("r");` -> **4**
- `"Universidad".lastIndexOf('i');` -> **7**
- `"Quilmes".substring(2,4);` -> **il**
- `(a.length() + a).startsWith("a");` -> **false**
- `s == a;` -> **true**
- `a.substring(1,3).equals("bc")` -> **true**

3. Tipos primitivos

Teniendo en cuenta la [documentación oficial de Java](#):

1. ¿Qué son los tipos de datos primitivos?

Son los tipos propios del lenguaje y no representan objetos sino el valor real que le estemos dando.

2. ¿Cuál es la diferencia entre un int y un Integer?

Un int es un tipo primitivo de java y Integer es la clase de ese tipo primitivo.

3. ¿Si se define una variable de instancia de tipo int cual sería su valor predeterminado? ¿Y si se define una de tipo Integer? Haga la prueba en Eclipse.

El valor de int como es primitiva toma el valor real. Y si se define como clase se guarda como instancia de ese Wrapper.

4. Responder la pregunta del punto anterior (3), pero ahora en lugar de definir una variable de instancia se define una variable de método.

El valor de int sería el real que le pases por parámetro. Y si se define en la clase Integer sería el valor referenciado.

4. Multioperador

Programa la clase `Multioperador`, que permite aplicar las operaciones de suma, resta y multiplicación

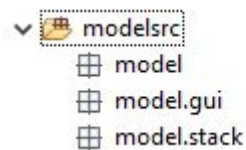
sobre [ArrayLists](#) de enteros. Es decir, poder sumar todos los números que contiene, poder restar todos los números que contiene y poder multiplicar a todos los números que contiene entre sí.

RTA: Repo -> <https://github.com/ManuelMichellon/unqui-po2-michellon>

5. Jerarquía de paquetes

Cree una clase que se encuentre en el paquete **model**, otra clase que se encuentre en el paquete **model.gui** y otra que se encuentre en el paquete **model.stack** . Compílelas utilizando la forma que crea conveniente.

1. ¿Cómo están organizadas en el sistema de archivos?



Los paquetes se verán de esta manera dentro de una src folder.

2. ¿Encuentra alguna relación entre el nombre del paquete y la ubicación de los archivos fuentes de las clases (.java) y los archivos compilados (.class)?

No se a que se refieren con archivos .class ya que todas las clases y testcase que se crearon en los paquetes son .java.

6. Point

Diseñe e implemente la clase `Point` (punto). La misma debe tener el siguiente comportamiento:

1. Debe ser posible crearse indicando como referencia los valores x e y
2. También debe ser posible crear un punto directamente sin enviarles parámetros, en este caso el punto debe crearse en las coordenadas (0,0).
3. Debe ser posible mover un punto a otra posición.
4. Sumarse con otro punto y como resultado obtener un nuevo punto con las valores de x e y sumados.

RTA: Repo -> <https://github.com/ManuelMichellon/unqui-po2-michellon>

7. Rectángulo

Utilizando el punto implementado anteriormente, defina el comportamiento de un `Rectángulo` definido en un espacio de dos dimensiones, es decir, poseer una ubicación en un espacio de coordenadas `x` e `y`. Los rectángulos deben tener el siguiente comportamiento:

1. Crearse en forma apropiada y asegurando su consistencia.
2. Obtener el área
3. Obtener el perímetro.
4. Determinar si son horizontales o verticales.

Además, diseñe la clase `Cuadrado` (re)utilizando lo anterior.

Realice los **test de unidad** correspondientes y luego implemente en java.

RTA: Repo -> <https://github.com/ManuelMichellon/unqui-po2-michellon>

8. Encapsulamiento

Implemente la clase `Persona` en Java. Una persona tiene un nombre y una fecha de nacimiento, por lo que debe ser posible pedirle su nombre, fecha de nacimiento y edad. En base a esto:

1. Responder: Si un objeto cualquiera que le pide la edad a una `Persona`: ¿Conoce cómo ésta calcula u obtiene tal valor? ¿Cómo se llama el mecanismo de abstracción que permite esto?

A través del acceso indirecto a las variables, en este caso `getter`.

2. Agregue a la clase `Persona` definida anteriormente el método `menorQue(Persona persona)` que recibe como parámetro a otra persona y retorna `true` en caso de que el receptor sea menor en edad que el parámetro, o `false` en caso contrario.
3. Agregue a la clase `Persona` un método (de clase) de creación, respetando la siguiente firma:
`Persona(String nombre, Date fechaNacimiento)`
que recibe como parámetros el nombre y la fecha de nacimiento de la persona a crear, crea una nueva instancia de `Persona` y la retorna inicializada con los valores recibidos como parámetro.

RTA: Repo -> <https://github.com/ManuelMichellon/unqui-po2-michellon>

9. Equipo de Trabajo

1. Defina la clase `Persona` y modélela en Java. Una persona tiene su *nombre*, *apellido* y *edad*.
2. Defina la clase `EquipoDeTrabajo` y modélela en Java. Un equipo tiene un *nombre* y un conjunto de *integrantes* (que son instancias de `Persona`).
3. Un `EquipoDeTrabajo` debe saber responder su nombre y el promedio de edad de sus integrantes.
4. Instancie un `EquipoDeTrabajo`, instancie 5 `Personas` y agreguelas al mismo.
5. Pida al `EquipoDeTrabajo` el promedio de edad de sus integrantes e imprima el resultado devuelto.

Régi par la licence [Creative Commons: Licence d'attribution en partage identique 4.0](https://creativecommons.org/licenses/by/4.0/)