* 1. ***TP N°1: Introducción a la POO***

|  |
| --- |
| La siguiente guía cubre los contenidos vistos en la clase teórica **1. Introducción a la POO** |

***Ejercicio 1***

Al elegir una bicicleta, están las de *small gear* (fáciles de pedalear pero no logran mucha velocidad) y las de *big gear* (más difícil de pedalear pero permite ir a más velocidad). El *gear* controla cuántas veces las ruedas completan un circuito completo con los pedales. Los términos *small* y *big* no son muy precisos. Es por eso que los ciclistas utilizan el término *ratio* para calcular el número de dientes del *gear*.

Los *ratios* pueden calcularse con el siguiente script de Java:

|  |
| --- |
| public class RatioScript {  public static void main(String[] args) {  int chainring = 52;  double cog = 11;  double ratio = chainring / cog;  System.out.println(ratio); // 4.7272727272727275  chainring = 30;  cog = 27;  ratio = chainring / cog;  System.out.println(ratio); // 1.1111111111111112  }    } |

Se desea crear una clase Gear que ofrezca una funcionalidad similar al del script. Para eso es necesario identificar las variables de instancia y los métodos.

**Diseñar el diagrama de clases correspondiente.** ¿Cómo cambiaría el programa de prueba con el uso de esta nueva clase?

***Ejercicio 2***

Se desea modelar a un número complejo (parte real y parte imaginaria).

La clase ComplexNumber debe permitir:

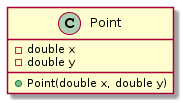
* Instanciar un objeto a partir de su parte real y su parte imaginaria
* Instanciar un objeto a partir de su parte real (asumiendo cero como parte imaginaria)
* Consultar su parte real y su parte imaginaria
* Que una instancia de ComplexNumber pueda sumarse con otra instancia de ComplexNumber, obteniendo así una nueva instancia con el resultado de la operación.

Identificar las variables y los métodos necesarios. ¿Cuántos métodos constructores se necesitan?

**Diseñar el diagrama de clases correspondiente.**

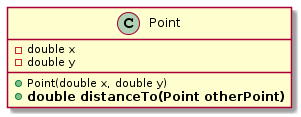
***Ejercicio 3***

Se cuenta con la clase Point que modela un punto en el plano 2D.

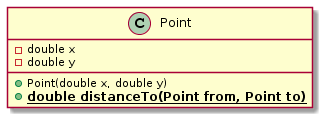
[](https://www.plantuml.com/plantuml/img/SoWkIImgAStDuU9ApaaiBbO8oC_CAr6evb800bs5bFpIf9nKXGfKRYMOgmrHgm5JemENrkIgvU9oICrB0PeB0000)

Se desea implementar un método que calcule la distancia entre dos puntos. **Indicar cuál de las siguientes opciones es la más apropiada desde el punto de vista del diseño orientado a objetos y por qué.**

1. Agregar un **método de instancia**

[](https://www.plantuml.com/plantuml/img/SoWkIImgAStDuU9ApaaiBbO8oC_CAr6evb800bs5bFpIf9nKXGfKRYMOgmrHgm5JemENreJAszhQmiHICe4MvIMdXkHhG6p8Byb8BG8pDRbgkRWSKlDIWAu00000)

1. Agregar un **método de clase**

[](https://www.plantuml.com/plantuml/img/SoWkIImgAStDuU9ApaaiBbO8oC_CAr6evb800bs5bFpIf9nKXGfKRYMOgmrHgm5JemENreJAszhQAbG3BIZ9JAw5IQPa0WNoabD3yZKWLgKLvUVgGAqjoTVagkNYSaZDIm4w2G00)

***Ejercicio 4***

Se desea modelar a un afiliado de una prepaga en la clase Affiliate. Cada afiliado cuenta con un nombre, apellido, fecha de nacimiento y un número de afiliado.

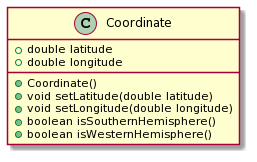
La clase debe contemplar además una numeración incremental única para todos los objetos de la misma, de forma que el primer objeto afiliado que se instancie tenga número 1000, el segundo afiliado que se instancie tenga número 1001 y así sucesivamente.

Para ello se cuenta con un método getAndIncrement() que será invocado en el constructor de Affiliate. Este método retorna en su nombre el número que le corresponde al nuevo afiliado que se está creando y actualiza el/los campo/s necesarios para la próxima creación. El método getAndIncrement() ¿es de instancia o de clase? ¿El número de afiliado es una variable de instancia o de clase? ¿Y el último número entregado?

**Diseñar el diagrama de clases correspondiente.**

***Ejercicio 5***

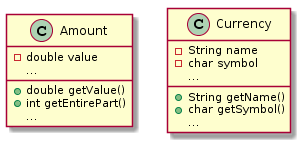
La clase Coordinate modela a una coordenada geográfica y está compuesta por una latitud y una longitud en grados decimales. La clase presenta métodos de instancia para setear ambos campos. Además ofrece métodos que indican si la coordenada está por debajo de la línea del Ecuador o si está al oeste del meridiano de Greenwich.

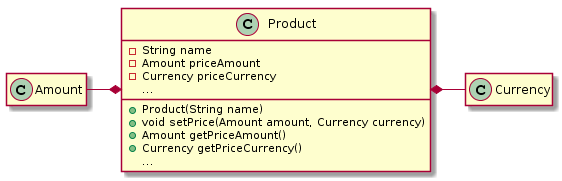
[**](https://www.plantuml.com/plantuml/img/POyn3i8m34Ltdy9Z49UenCB0ni3i4ej6IcqKEoo8krEaLgc29-lf-NzvCAVYTSu0zqncU58jCGavunjmcID6hMDcpEJ9Q-IEgZpskDs7mufUcY8Q-sKD25tWdxT5XhvZCqVLp2IOxAhL9ovovZdPSzbQRPDkRDux7u219Vu-_m80)

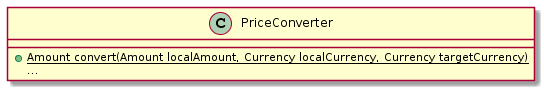
**Analizar el diseño indicando las mejoras que se podrían hacer.**

***Ejercicio 6***

La clase Amount modela a un importe monetario y la clase Currency modela a la moneda del importe. Estas clases son utilizadas por métodos de la clase Product que permiten setear el precio del mismo y por un método utilitario de la clase Converter que permite convertir un monto de cierta moneda a otra.

[](https://www.plantuml.com/plantuml/img/RO_F2i8m38VlVOeS5N7FC16l8WoyZguBiz0_aAQ38RwxMUa6Wha4lkQNhsaJQ-BidLB6wPJWucCE36y5KcSOOnuSmgnTneAQfYczxqixRHCJydCTEXmhjx95u2sm9No8fBwi2pwRxPg9C9XbztLCDamGjCTAp4iJfCKFqVt_G8s8wowftLzo0hiI_PMt64Qv-Gi0)

[******](https://www.plantuml.com/plantuml/img/JL1R3eCW4FpVK_1f7te6crx0f0Sm5BP28j30qgPfUlSgWihFxCxix99f0t5FqOm0OkGXiCux6GMnBx3frUn6Njk1MMumCnVZeYNsz5hWKcVX6hr7ApwBLBeaDasJiEyJdCgPQhV-aAMNqv85f6xUKkLhFC5vko5oKLnvRiY-fQsAkjgANeWyCV_l1w2qH6RGtD472CgzrpWCad8o01JYM8_ue9B6ZlPwKBI50T2YbLF0Vm00)

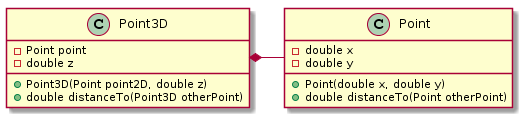
[******](https://www.plantuml.com/plantuml/img/JOxH2O0m34J_CqK-5QKpA2xW1b9Y4A6saAQ2YBjRh8Btatl7mQMBQaNJvW38sHXnb9Lu27vdKHOy0REcwJaDdhckArtORo5vHIh5waKNoBhYMnoI27iwIlhHBy_B2-k7zJDXZ84Be6C_vwTk)

**Analizar el diseño indicando las mejoras que se podrían hacer.**

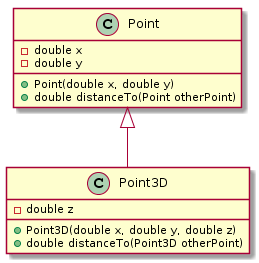
***Ejercicio 7***

Se desea diseñar la clase Point3D para representar puntos en el espacio (x, y, z). **Indicar cuál de las siguientes opciones es la más apropiada desde el punto de vista del diseño orientado a objetos y por qué.**

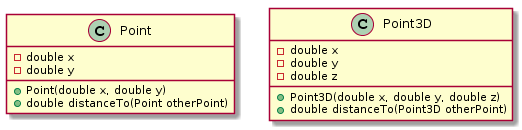
1. Composición: Point3D cuenta con una variable de instancia Point y una variable de instancia z

[](https://www.plantuml.com/plantuml/img/SoWkIImgAStDuU9ApaaiBbO8oC_CAr6evb800bs5bFpIf9nKXGfKRYMOgmrHgm5JemENreJAG_afcK1Byf9JG_8r8ARdbsIa5e6PcborA1ORky2jXYWj096ejbSXssxiee6aqCW5xeWg0eu0seJc39YmbcvHPdf62TG5N5mEgNafm0020000)

1. Herencia: Point3D es subclase de Point y cuenta únicamente con la variable de instancia z (ya que hereda x e y se heredan de la clase Point).

[](https://www.plantuml.com/plantuml/img/SoWkIImgAStDuU9ApaaiBbO8oC_CAr6evb800bs5bFpIf9nKXGfKRYMOgmrHgm5JemENreJAG_afcK1Byf9JG_8r8ARdbsIa5e6PcborA1ORkw1RNOLibx4B5jlWh2e2zW8DHxCPuXoR6brTcFbSN0wfUId0y000)

1. Sin Relación: Point3D define tres variables de instancia x, y, z.

[](https://www.plantuml.com/plantuml/img/SoWkIImgAStDuU9ApaaiBbO8oC_CAr6evb800bs5bFpIf9nKXGfKRYMOgmrHgm5JemENreJAG_afcK1Byf9JG_8r8ARdbsIa5e6PcborA1ORkn1bDPnRXUmIOnSiRe6pgWYu2cWrchiSKlDIW2430000)

Analizar el comportamiento de las clases en cada caso. Sin implementarlo, ¿cómo se calcula la distancia entre dos puntos en tres dimensiones? ¿Cómo se relaciona ese cálculo con el cálculo de la distancia en dos dimensiones?

***Ejercicio 8***

Se desea diseñar un conjunto de clases que modelan a figuras geométricas en el espacio 2D. Utilice una o más instancias de la clase Point para definir a una figura.

Diseñar primero las clases Rectangle y Triangle. Ambos deben contar con métodos para calcular perímetro y área. El rectángulo debe contar además con métodos que calculen base y altura. ¿Corresponde crear otra clase más?

Agregar luego las clases Circle y Ellipse como otras dos figuras más de la familia de clases, definiendo para ellas sólo el perímetro y el área. ¿Debería haber alguna relación entre ambas?

**Diseñar el diagrama de clases correspondiente.**

***Ejercicio 9***

Se desea agregar un método que determine si una instancia de la clase Point pertenece o no al contorno de una figura. ¿Dónde conviene definirlo? Pensar distintas alternativas y quedarse con la mejor desde el punto de vista POO.

**Actualizar el diagrama de clases del ejercicio anterior.**

***Ejercicio 10***

Se desea diseñar un conjunto de clases que modelan las cuentas de un banco, sobre las cuales se permiten hacer únicamente depósitos y extracciones. Todas las cuentas tienen un número identificador y un saldo (balance).

Existen dos tipos de cuentas: caja de ahorro y cuenta corriente. La cuenta corriente cuenta con un descubierto y la caja de ahorros no. En otras palabras, un monto puede retirarse de una caja de ahorro sólo si cuenta con los fondos. La cuenta corriente puede tener un saldo negativo, nunca mayor al monto del descubierto.

**Diseñar el diagrama de clases correspondiente.**