

Trabajo Práctico N° 03

Tema: Arrays – String – Herencia - Composicion

1. La clase Fechas es una lista de cadenas que representan fechas en siguiente formato: 1 de enero de 2003. Escribir métodos para:

- Retornar la cantidad de fechas correspondientes a un mes determinado.
- Hallar la cantidad de fechas correspondientes a la primera quincena.
- Retornar la cantidad de fechas

2. La clase Curso se compone del dni de los alumnos y sus notas respectivas. Es necesario realizar cálculos como:

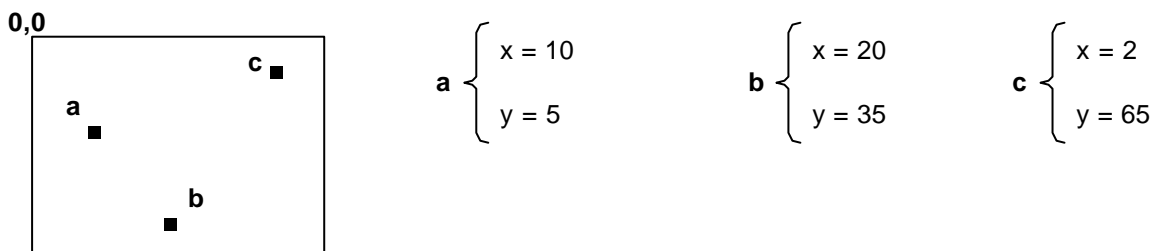
- Retornar el dni del alumno con mayor nota.
- Retornar el dni del alumno con menor nota.
- Retornar la nota promedio.
- Retornar la cantidad de aplazos.
- Retornar la nota de un determinado alumno.

Realizar dos implementaciones para la clase Curso:

- con dos arreglos, uno para los dni y otro para las notas,
- con un arreglo de objetos de la clase Alumno.

3. Un quiosco vende todo tipo de artículos. Un artículo tiene una descripción, un precio, un código, y un stock (cantidad de unidades disponibles del artículo). El quiosco vende y compra artículos, operaciones que decrementan e incrementan el stock. El quiosco vende y compra cuando está abierto. Periódicamente el quiosco realiza control de stock para ello controla los artículos que están sin stock.

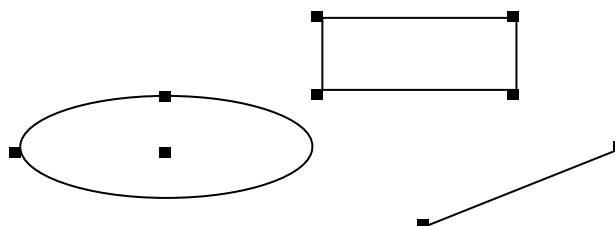
4. Un punto se puede implementar mediante sus dos coordenadas x e y. Teniendo en cuenta que en Java el sistema de coordenadas comienza en el extremo izquierdo superior en $x = 0$ e $y = 0$. x avanza hacia abajo (filas) e y avanza hacia la derecha (columnas). Por ejemplo:



Codificar la clase Punto.

5. Teniendo en cuenta las siguientes composiciones:

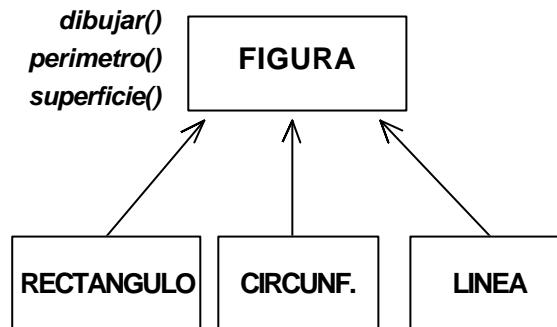
- una línea "tiene" dos puntos
- un rectángulo "tiene" 4 puntos
- una circunferencia "tiene" 3 puntos



Trabajo Práctico N° 03

- Codificar las clases Linea, Rectángulo y Circunferencia (deben incluir métodos que calculen sus superficies y perímetros, setter y getter).
- Escribir un programa en el que se creen objetos de tipo Linea, Rectángulo y Elipse. Probar los métodos codificados.

6. Codificar la siguiente jerarquía de herencia: (trabajar con la clase punto)



Tener en cuenta las siguientes propiedades:

- Cada figura tiene un procedimiento para su dibujo
- a una figura se le puede calcular el perímetro
- a una figura se le puede calcular la superficie
- una figura tiene un color de fondo
- un figura tiene un color de borde

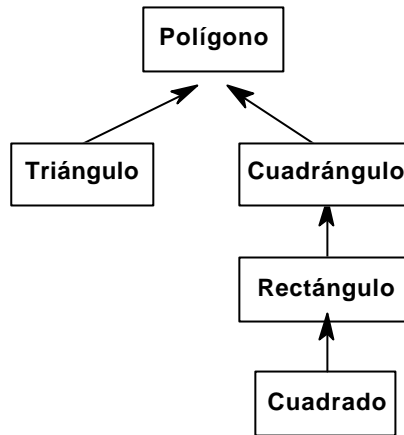
7. Agregar los siguientes métodos a Figura y a sus subclases (debe usar los parámetros que necesite):

- moverIzquierda()
- moverDerecha()
- moverArriba()
- moverAbajo()
- pintarFondo()
- pintarBorde()

8. Con las clases de los ejercicios anteriores implemente los ejemplos 1 y 2 de polimorfismo del apunte, en forma completa.

9. Teniendo en cuenta la siguiente jerarquía de clases:

Trabajo Práctico Nº 03



Y las declaraciones:

Triangulo t;
Polígono p;
Cuadrángulo c;
Rectángulo r;
Cuadrado cd;

Indique cuales de las siguientes sentencias no son validas y por qué:

<p>p = c; p = r; cd = c; c = t;</p>	<p>p = r; r = p; p.diagonal(); // diagonal es una propiedad de los cuadrángulos t.perímetro(); // perímetro es una propiedad de los polígonos</p>
---	---

10. Explique claramente cual es el error en las siguientes jerarquía de herencia implementada en Java:

<pre>public class Uno { public int f() { return k();} public int k() { return 50;} }</pre>	<pre>public class Dos extends Uno { public int f() { return super.f() + h();} public int h() { return 20;} }</pre>	<pre>public class Tres extends Dos { public int f() { return 80;} public int k() { return super.f() + 10;} }</pre>
---	---	---

11. Dada la siguiente definición de clases:

<pre>public class AA { public int g() { return 5;} }</pre>	<pre>public class BB extends AA { public int g() { return 3;} public int h() { return g();} }</pre>	<pre>public class CC extends BB { public int g() { return super.g();} public int h() { return super.h() + 1;} }</pre>
--	--	--

Y las declaraciones:

AA p; BB q; CC r; int M;

Trabajo Práctico Nº 03

Haga la **traza** para cada segmento de código. Si encuentra una sentencia errónea indique si el error es en tiempo de ejecución o de compilación. De haber error indique, además el valor de M hasta la última sentencia válida.

(a)	(b)	(c)
<pre>p = q; q = r; M = p.g(); M = M + p.h();</pre>	<pre>p = q; M = r.f(); q = r; M = M + q.h();</pre>	<pre>p = q; M = q.g() + p.g() + r.f(); q = r; r = null; M = q.h();</pre>

12. Dada la siguiente definición de clases:

<pre>public class Uno { public int f() { return g()+10;} public int g() { return 2;} }</pre>	<pre>public class Dos extends Uno { public int g() { return 50 + k();} public int k() { return 100;} }</pre>
---	---

Y las declaraciones:

Uno a, b; Dos c, d; int n,m;

Indique para cada segmento de código si existen sentencias incorrectas y si el error es en tiempo de ejecución o de compilación, justificando su respuesta. Siempre que sea posible, señale además los valores finales de n y m.

13. En una biblioteca virtual se pueden consultar publicaciones. Estas pueden ser libros y/o artículos de revistas. Los libros se pueden buscar por autor, título o isbn(código único de libro), los artículos solo por autor o título. Al ubicarse una publicación se accede a todos los datos de la misma. Los libros poseen una editorial y cantidad de páginas, los artículos poseen cantidad de páginas y revista en la que se publicó.

- Dibujar el diagrama de clases y los diagramas de secuencia que considere necesarios.
- Implementar la registración, búsqueda y consulta de publicaciones en la biblioteca virtual.

14. Sobrecargar el constructor de la clase Hora:

- Hora() // sin parámetros, se da un valor a hora, minuto y segundo por defecto
- Hora(int s, int m, int s) // se pasan los parámetros correspondientes a hora, minuto, segundo.

15. Un rectángulo se representa por sus 4 puntos: superior izquierdo (si), superior derecho (sd), inferior izquierdo (ii) e inferior derecho (id). Por tanto al crearse un objeto de esta clase se inicializan tales puntos. Escribir la clase Rectángulo, sobrecargando el constructor ya que se puede construir a partir de distinto tipo de información, como ser:

- i. Rectángulo(.....) // se pasan como parámetros los 4 puntos
- ii. Rectángulo(.....) // se pasan como parámetros el punto si, el ancho y largo de rectángulo
- iii. Rectángulo(.....) // se pasan como parámetros los puntos si e id.