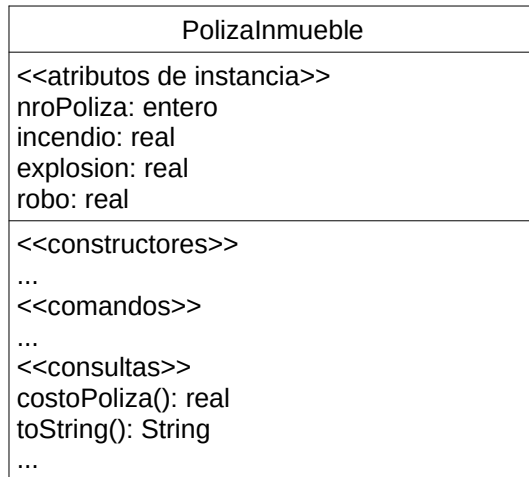




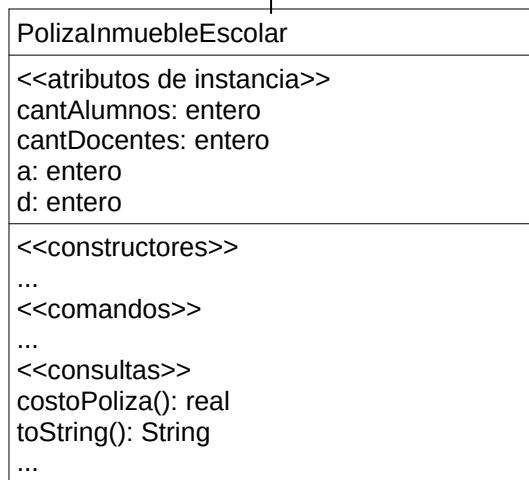
PRACTICO N° 12 Herencia y Polimorfismo

EJERCICIO 1. Implemente en Java el diagrama de clases que sigue, incluyendo todos los métodos triviales y considerando que la forma de póliza más simple es la póliza de inmueble y dentro de ella se distinguen las pólizas de inmuebles escolares.



costoPoliza()

El costo de la póliza por un inmueble es la suma de las coberturas por robo, explosión e incendio.



CostoPoliza()

Es el costo de cualquier póliza más a pesos por cada alumno y d pesos por cada docente.



Introducción a la Programación Orientada a Objetos

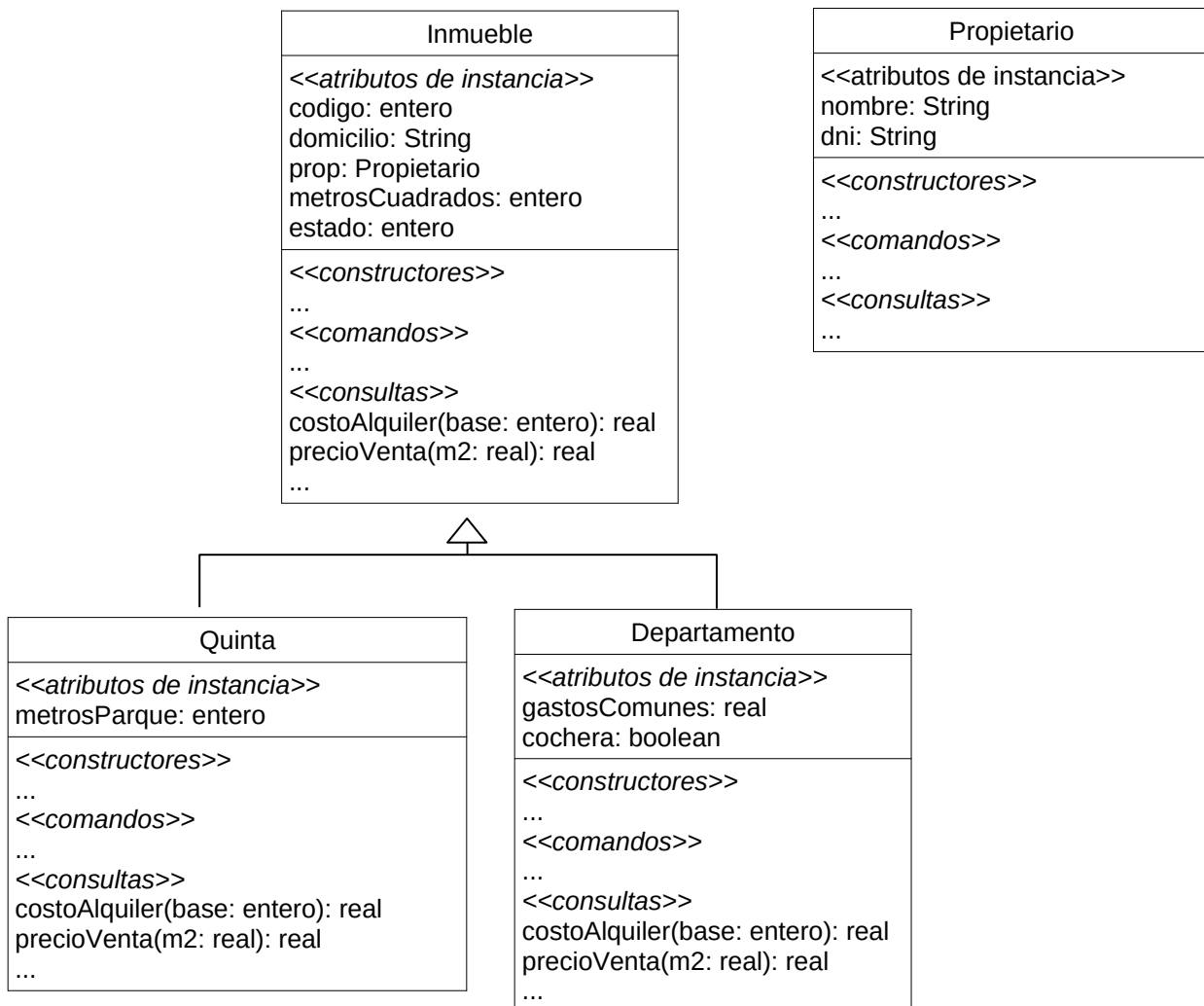
DCIC - UNS

2019



EJERCICIO 2. Implemente en Java el diagrama de clases que sigue incluyendo todos los métodos triviales.

- El costo del alquiler de cualquier inmueble se calcula a partir del valor base recibido como parámetro más \$100 por metro cuadrado de superficie del inmueble. El costo del alquiler de una quinta es el de cualquier inmueble más \$500 por cada 15 metros cuadrados de parque y el de los departamentos es el costo de cualquier inmueble más \$2000 si tiene cochera.
- El precio de venta de cualquier inmueble se calcula como cantidad de metros cuadrados por el valor del metro cuadrado, el cual es recibido por parámetro. En el caso de las quintas cada metro cuadrado de parque es igual a la mitad del valor del metro cuadrado construido. El precio de venta de los departamentos, suma al costo del inmueble el equivalente a 3 metros cuadrados en caso de tener cochera.





Introducción a la Programación Orientada a Objetos

DCIC - UNS

2019



EJERCICIO 3. Dadas las siguientes definiciones:

```
class Uno {
    int f () { return 0; }
    int g () { return 1; }
    int h () { return 2; }}
class Dos extends Uno {
    int f (int x) { return 10*x; }
    int g () { return 11; }}
class Tres extends Dos {
    int f () { return 20; }
    int l () { return 21; }
    int h () { return 12; } }
```

```
Uno u1,u2,u3,u4;
Dos d1,d2;
Tres t;
```

```
t = new Tres();
u1 = new Uno();
u2 = new Dos();
u3 = new Tres();
u4 = t;
d1 = new Dos();
d2 = new Tres();
```

Muestre la salida de las siguientes instrucciones:

<pre>System.out.println (u1.f()); System.out.println (u2.f()); System.out.println (u3.f()); System.out.println (u4.f()); System.out.println (u1.g()); System.out.println (u2.g()); System.out.println (u3.g()); System.out.println (u4.g()); System.out.println (u1.h()); System.out.println (u2.h()); System.out.println (u3.h()); System.out.println (u4.h());</pre>	<pre>System.out.println (d1.f()); System.out.println (d2.f()); System.out.println (d1.f(-1)); System.out.println (d2.f(-1)); System.out.println (d1.g()); System.out.println (d2.g()); System.out.println (d1.h()); System.out.println (d2.h());</pre>	<pre>System.out.println (t.f()); System.out.println (t.g()); System.out.println (t.h()); System.out.println (t.f(-1)); System.out.println (t.l());</pre>
--	--	--



Introducción a la Programación Orientada a Objetos

DCIC - UNS

2019



EJERCICIO 4. Dadas las siguientes definiciones de clases y las declaraciones de variables

```
class Uno {
    int f () { return 1; }
    int g () { return 2; }
    int h () { return 0; }}
class Dos extends Uno {
    int f (int x) { return x; }
    int g () { return f()+10; }
    int h () { return super.h()+100; }}
class Tres extends Dos {
    int f () { return 33; }
    int l () { return 44; }}
```

```
Uno u1,u2,u3,u4;
Dos d1,d2,d3;
Tres t1,t2,t3;
```

Indique cuáles de las siguientes instrucciones provocarán errores de compilación y de ejecución. Siempre que sea posible, muestre cuál sería la salida por consola.

- (1) t3 = new Tres();
- (2) u1 = new Uno();
- (3) u2 = new Dos();
- (4) u3 = new Tres();
- (5) u4 = t3;
- (6) d1 = new Uno();
- (7) d2 = new Dos();
- (8) d3 = new Tres();
- (9) t1 = (Tres) u2;
- (10) t2 = (Tres) d3;

System.out.println (u1.f()); System.out.println (u2.f()); System.out.println (u3.f()); System.out.println (u4.f()); System.out.println (d1.f()); System.out.println (d2.f()); System.out.println (d3.f()); System.out.println (t1.f()); System.out.println (t2.f()); System.out.println (t3.f()); System.out.println (u1.f(11)); System.out.println (u2.f(11)); System.out.println (u3.f(11)); System.out.println (u4.f(11)); System.out.println (d1.f(11)); System.out.println (d2.f(11)); System.out.println (d3.f(11));	System.out.println (u1.g()); System.out.println (u2.g()); System.out.println (u3.g()); System.out.println (u4.g()); System.out.println (d1.g()); System.out.println (d2.g()); System.out.println (d3.g()); System.out.println (t1.g()); System.out.println (t2.g()); System.out.println (t3.g()); System.out.println (u1.h()); System.out.println (u2.h()); System.out.println (u3.h()); System.out.println (u4.h()); System.out.println (d1.h()); System.out.println (d2.h()); System.out.println (d3.h());	System.out.println (u1.l()); System.out.println (u2.l()); System.out.println (u3.l()); System.out.println (u4.l()); System.out.println (d1.l()); System.out.println (d2.l()); System.out.println (d3.l()); System.out.println (t1.f(11)); System.out.println (t2.f(11)); System.out.println (t3.f(11)); System.out.println (t1.h()); System.out.println (t2.h()); System.out.println (t3.h()); System.out.println (t1.l()); System.out.println (t2.l()); System.out.println (t3.l());
--	--	--



Introducción a la Programación Orientada a Objetos

DCIC - UNS

2019



EJERCICIO 5. Dadas las siguientes definiciones:

<pre>class Alfa{ protected int n; protected String s; public Alfa(){ n= 10; s= new String ("Alfa"); } public String p(int x){ return s+" p n: "+n+" x: "+x; } public String q(int x){ return s+" q n: "+n+" x: "+x; } public String r(int x){ return s+" r n: "+n+" x: "+x; } public String v(char x){ return s+" v n: "+n+" x: "+x; } }</pre>	<pre>class Beta extends Alfa{ protected int m; protected String ss; public Beta(){ m= 11; ss= new String ("Beta"); } public String p(int x){ return ss+" p m: "+m+" x: "+x; } public String q(String x){ return ss+" q m: "+m+" x: "+x; } }</pre>	<pre>class Delta extends Beta{ protected int l; protected char ch; public Delta(){ l= 12; ch = 'D'; } public String q(int x){ return ch+" q l: "+l+" x: "+x; } public String t(int x){ return ch+" t l: "+l+" x: "+x; } public String v(int x){ String sss= ch+" v l: "+l+" x: "+x; return sss; } }</pre>
--	---	---

Y dadas las siguientes declaraciones de variables:

Alfa a1 = new Alfa();

Alfa a2 = new Beta();

Beta a3 = new Beta();

Alfa a4 = new Delta();

Beta a5 = new Delta();

Delta a6 = new Delta();

Indique cuáles de las siguientes instrucciones provocarán errores de compilación y de ejecución. Siempre que sea posible, muestre cuál sería la salida por consola.

<pre>System.out.println(a1.p(5)); System.out.println(a2.p(5)); System.out.println(a3.p(5)); System.out.println(a4.p(5)); System.out.println(a5.p(5)); System.out.println(a6.p(5));</pre>	<pre>System.out.println(a1.q(6)); System.out.println(a2.q(6)); System.out.println(a3.q(6)); System.out.println(a4.q(6)); System.out.println(a5.q(6)); System.out.println(a6.q(6));</pre>	<pre>System.out.println(a1.r(7)); System.out.println(a2.r(7)); System.out.println(a3.r(7)); System.out.println(a4.r(7)); System.out.println(a5.r(7)); System.out.println(a6.r(7));</pre>
<pre>System.out.println(a1.q("q8")); System.out.println(a2.q("q8")); System.out.println(a3.q("q8")); System.out.println(a4.q("q8")); System.out.println(a5.q("q8")); System.out.println(a6.q("q8"));</pre>	<pre>System.out.println(a1.t(9)); System.out.println(a2.t(9)); System.out.println(a3.t(9)); System.out.println(a4.t(9)); System.out.println(a5.t(9)); System.out.println(a6.t(9));</pre>	<pre>System.out.println(a1.v(7)); System.out.println(a2.v(7)); System.out.println(a3.v(7)); System.out.println(a4.v(7)); System.out.println(a5.v(7)); System.out.println(a6.v(7));</pre>