Programación de Sistemas
Grado en Ingeniería de Tecnologías de Telecomunicación

Leganés, 13 de marzo de 2018 — Examen parcial 1 (teoría)
Duración de la prueba: 20 min — Puntuación: 3 puntos sobre 10 del examen

Sólo una opción es correcta en cada pregunta. Cada respuesta correcta suma 0,3 *puntos. Cada respuesta incorrecta resta* 0,1 *puntos. Las preguntas no contestadas no suman ni restan puntos.*

- Marca la respuesta a cada pregunta con una equis ("X") en la tabla de abajo.
- Si marcas más de una opción o ninguna opción, la pregunta se considera no contestada.
- Rellena **tus datos personales** antes de comenzar a realizar el examen.

Nombre:	Grupo:

Firma:

NIA:

	A	B	C	D		A	B	C	D
1					6				
2					7				
3					8				
4					9				
5					10				

1.- Dado el siguiente código, ¿qué se imprime por pantalla?

```
public class P {
    private int[] array;
    public static void main(String[] args){
        P p = new P();
        for (int i = 0; i < p.array.length; i++){
            System.out.println(p.array[i]);
        }
    }
}
```

(a) *** Hay un error y se lanza una *NullPointerException*
(b) Hay un error y se lanza una *ArrayIndexOutOfBoundsException*
(c) 0
(d) null

2.- Dado el siguiente código, ¿qué se imprime por pantalla?

```
public class P {
    private static int p1 = 0;
    private static int p2 = 1;
    public P(){
        p2++;
        p1=p2;
    }
    public static void main(String[] args){
        P a = new P();
        P b = new P();
        System.out.println(p1+p2);
    }
}
```

(a) *** 6
(b) 2
(c) 3
(d) 4

3.- Completa la parte del código marcada con puntos suspensivos

```
public class P {
    public int divide (int a, int b) ... {
        if (b == 0){
            throw new Exception();
        } else
            return a/b;
    }
}
```

(a) *** *throws Exception*

(b) *throw new Exception()*

(c) *throw Exception*

(d) *throws new Exception()*

4.- Los métodos definidos con el modificador *final*

(a) *** No pueden ser sobreescritos

(b) No pueden devolver *void*

(c) No pueden ser sobrecargados

(d) No pueden ser privados

5.- Dadas las clases *C1*, *C2* y *C3* y las interfaces *I1* e *I2* ¿Cuál de las siguientes sentencias es correcta?

(a) *** public interface I1 extends I2

(b) public interface C1 implements I1, I2

(c) public class C1 extends C2, C3

(d) public class C1 implements I1, I2 extends C2

6.- En la sobreescritura de métodos hay:

(a) *** métodos con mismo nombre, número y tipo de parámetros

(b) métodos con distinto nombre, pero mismo número y tipo de parámetros

(c) métodos con mismo nombre, pero distinto número y/o tipo de parámetros

(d) métodos con distinto nombre, número y tipo de parámetros

7.- Dada la clase A, su clase hija abstracta B, y la clase hija de esta última C no abstracta, todas ellas con constructores que no reciben parámetros, señala la sentencia que produce un error en tiempo de compilación:

(a) *** C c = (C) new B();

(b) A a = new C();

(c) C c = (C) new A();

(d) B b = new C();

8.- Dada la clase no abstracta A, la clase abstracta B que hereda de A y tiene el método abstracto b, y la clase no abstracta C que hereda de B, podemos decir que:

(a) *** Solamente podemos crear objetos de A y C

(b) A y C deben implementar el método abstracto b

(c) Aes una clase derivada de C

(d) En un array de tipo B ($B[]$) podemos almacenar objetos de A, B y C

9.- Dado el siguiente código y su clase de test, ¿qué cobertura de métodos se alcanza?

```
public class A {
    private int a;

    public A(int a) {
        this.a = a;
    }

    public boolean a() {
        if (a <= 0) {
            return false;
        } else if (a > 100) {
            return false;
        } else
            return true;
    }
}

public class ATest {
    A a;
    @Before
    public void setUp() throws Exception {
        a = new A(10);
    }

    @Test
    public void testA() {
        assertTrue(a.a());
    }
}
```

(a) *** 100 %
(b) 25 %
(c) 50 %
(d) 66 %

10.- Se dispone del método *double acos(double a)* que calcula la el arcocoseno del valor recibido como parámetro, y se desea hacer una prueba de caja negra. De las siguientes opciones, selecciona aquellos valores que permite cubrir todas las clases de equivalencia de este método. Pista: recuerda que el método que calcula el coseno de un ángulo entre 0 y 2*PI devuelve un valor en el rango entre -1 y 1 (incluidos)

(a) *** -2.5, 0.0, 2.5
(b) 0.0, 1.0, 2.5
(c) -1.0, 0.0, -1.0
(d) -2.0, -0.5, 0.5