

SI-Pass 2tinyurl.com/sipass2oopf**1)**

Skriv kod för en Factory. Factoryn ska returnera en instans av ett djur. För att åstadkomma detta ska Factoryn ta emot en int som parameter för antalet ben och utifrån detta värde returnera något av följande djur:

- Anka
- Hund
- Spindel

Om antalet ben inte överensstämmer med något djur ovan ska ett Monster returneras.

2)

Skriv kod som skapar en array med tre olika djur i. Varje djur ska skapas med hjälp av Factoryn ni just skrev. Arrayen måste utnyttja polymorfism.

(Tips: Antag att alla djuren extender en Animal-klass)

3)

Betrakta kodraderna:

```
Animal animal1 = new Cat();  
Object list1 = new List();
```

a)

Vilka är de statiska typerna?

b)

Vilka är de dynamiska typerna?

c)

Vilken/vilka av typerna kontrolleras under kompileringen av programmet?

d)

Vilken/vilka av typerna kontrolleras under exekveringen av programmet?

SI-Pass 2

tinyurl.com/sipass2oopf

4)

Betrakta nedanstående klasser och interface:

```

public interface A {
    public void a();
} //A

public interface B {
    public void b();
} //B

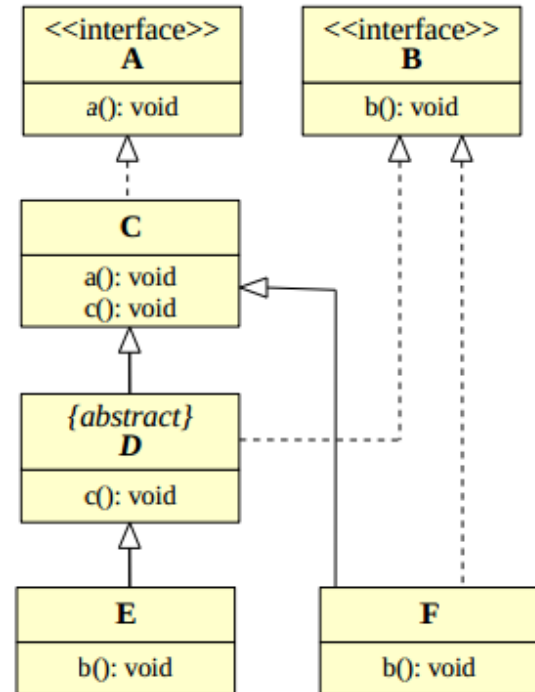
public class C implements A {
    public void a() {
        System.out.println( "a() in C" );
    } //a
    public void c() {
        System.out.println( "c() in C" );
    } //c
} //C

public abstract class D extends C implements B {
    public void c() {
        System.out.println( "c() in D" );
    } //c
} //D

public class E extends D {
    public void b() {
        System.out.println( "b() in E" );
    } //b
} //E

public class F extends C implements B {
    public void b() {
        System.out.println( "b() in F" );
    } //b
} //F

```



Vad blir resultatet för var och en av följande satser (ger kompileringsfel, ger exekveringsfel, skriver ut xxx, etc)?

- | | |
|---|---|
| a) C x = new E(); x.c(); | e) C x = new F(); x.b(); |
| b) Object o = new E(); System.out.println(o instanceof D); | f) B x = new F(); D y = (D) x; x.b(); |
| c) C x = new D(); x.c(); | g) D x = new C(); x.a(); |
| d) B x = new E(); D y = (D) x; y.b(); | h) A x = new F(); C y = x; y.a(); |

SI-Pass 2

tinyurl.com/sipass2oopf**5)**

Betrakta följande klasser:

```
public class Cat{  
    //  
}  
  
public class CatWithBoots extends Cat{  
    //  
}  
  
public class CatWithHat extends Cat{  
    //  
}
```

Med dessa klasser kan vi skapa en katt, en katt med stövlar samt en katt med hatt. Men hur gör vi om vi vill skapa en katt som har både stövlar och hatt?

a)

Kan vi skapa en ny CatWithBootsAndHat-klass som extendar både stövel- och hatt-klassen? Varför/Varför inte?

b)

Lös problemet genom att använda gränssnitt (interface).

c)

Problemet med interfaces är att de inte säger något om implementationen, dvs vad koden faktiskt ska göra. Detta går att råda bot på genom att använda delegering.

- Vad innebär delegering i Java?
- Lös problemet genom att använda delegering.

6)

Vad är skillnaden på preconditions och postconditions?

7)

Försök komma på olika sidoeffekter som kan inträffa i Java-program. Hur kan man undvika dessa?