BÀI TẬP 2

Trả lời câu hỏi

Lớp trừu tượng khác lớp bình thường ở điểm:

A. Không được viết code cho các phương thức bên trong
B. Phải khởi tạo đối tượng
C. Không được khởi tạo đối tượng
D. Không được có thuộc tính

Trả lời câu hỏi

Lớp trừu tượng có thể có:

A. Phương thức trừu tượng
B. Phương thức không trừu tượng
C. Cả hai
D. Không cái nào

Trả lời câu hỏi

Lớp trừu tượng Animal có hai lớp con Dog và Cat. Các hàm tạo đều không nhận tham số. Câu lệnh nào sau đây có thể thực hiện:

A. Animal a = new Animal();

B. Dog a = new Dog();

C. Cat a = new Animal();

D. Hai trong số 3 phương án trên

Trả lời câu hỏi

Khẳng định nào sau đây là đúng

A. Đối tượng thuộc lớp cha cũng là một đối tượng thuộc lớp con
B. Lớp cha có thể chứa phương thức trừu tượng
C. Có thể khởi tạo đối tượng của lớp trừu tượng bằng từ khoá new
D. Một lớp trừu tượng không thể chứa phương thức trừu tượng

Trả lời câu hỏi

Bạn ……. ghi đè phương thức toString() trong mỗi lớp mà bạn viết

A. Không thể
B. Có thể
C. Bắt buộc
D. Phải sử dụng interface StringListender để có thể

Trả lời câu hỏi

Phương thức equals() của lớp Object có bao nhiêu tham số

A. Không
B. Một
C. Hai
D. Bao nhiêu cũng được

Trả lời câu hỏi

Giả sử câu lệnh sau: thing.equals(anotherThing)

Trả lại giá trị là ĐÚNG

Chúng ta có thể khẳng định được:

A. thing là một đối tượng thuộc lớp Object
B. anotherThing có cùng kiểu với thing
C. Tất cả các trường trong anotherThing có cùng giá trị với các trường trong thing
D. Tất cả các phương án trên

Trả lời câu hỏi

Hàm equals() mặc định của lớp Object trả lại giá trị đúng khi hai đối tượng có cùng:

A. Giá trị trong tất cả các thuộc tính
B. Giá trị trong một thuộc tính bất kì
C. Kiểu
D. Địa chỉ bộ nhớ

Trả lời câu hỏi

Một lớp trong Java có thể kế thừa

A. Một lớp cha
B. Hai lớp cha
C. Không lớp cha
D. Bao nhiêu lớp cha cũng được

Trả lời câu hỏi

Để thực hiện đa kế thừa trong java chúng ta phải sử dụng:

A. Siêu lớp
B. Lớp trừu tượng
C. Interface
D. Không có cách nào

Trả lời câu hỏi

Có thể khởi tạo đối tượng thuộc:

A. Interface
B. Lớp trừu tượng
C. Cả hai
D. Không cái nào

Trả lời câu hỏi

Khi chúng ta có một lớp cha có chứa thuộc tính, phương thức cụ thể và một vài phương thức trừu tượng mà lớp con phải ghi đè, chúng ta nên dùng:

A. Lớp cụ thể

B. Lớp trừu tượng

C. Interface

D. Có thể dùng cả 3 cách trên

Cho biết kết quả đoạn mã sau

```
1  class Test {
2      int i;
3      Test(int i) {
4          this.i = 12;
5          i = 15;
6      }
7      void i() {
8          Test i = new Test(3);
9          System.out.println(i.i);
10         i.i();
11     }
12     public static void main(String[] toto) {
13         Test i = new Test(34);
14         i.i();
15     }
16 }
```

Cho biết kết quả đoạn mã sau

```
1  class C1 {
2      C2 c;
3      int i;
4      C1(int i, C2 c) {
5          if (c == null) this.c = new C2(i + 1, this);
6          else this.c = c;
7          this.i = i;
8      }
9      public static void main(String[] toto) {
10         C1 c = new C1(1, null);
11         C2 d = new C2(c.i, c);
12         C1 e = new C1(d.i, d);
13         System.out.println(e.i + "," + e.c.i + "," + e.c.c.i + "," + e.c.c.c.i);
14     }
15 }
16 class C2 {
17     C1 c;
18     int i;
19     C2(int i, C1 c) {
20         if (c == null) this.c = new C1(i + 1, this);
21         else this.c = c;
22         this.i = i;
23     }
24 }
```

Cho biết kết quả đoạn mã sau

```java
class B {
    int i = 5;
    B() {
        this.i = this.i - 1;
    }
    B(int i) {
        this();
        this.i = i;
    }
}
class C extends B {
    C(int i) {
        this.i = i;
    }
    public static void main(String[] arggggghhhh) {
        B b = new B(2);
        C c = new C(1);
        System.out.println(b.i + " " + c.i);
    }
}
```

Cho biết kết quả đoạn mã sau

```
1  class A {
2      A a;
3      A() {
4          this.a = this;
5      }
6      A(A a) {
7          this.a = a;
8      }
9      void m() {
10         if (this == this.a) System.out.println("Ahah!");
11         else System.out.println("Héhé!");
12     }
13 }
14 class B extends A {
15     B o;
16     B() {
17         super();
18         this.a = this;
19         this.o = (B) this;
20     }
21     void m() {
22         System.out.println("Ohoh!");
23     }
24     public static void main(String[] toto) {
25         A u = new A();
26         A i = new A(u);
27         A b = new B();
28         u.m();
29         i.m();
30         b.m();
31         ((B) b).o.m();
32     }
33 }
```

```java
interface Titi {
    int m1();
}

interface Tutu extends Titi {
    int m1();
    int m2();
}

abstract class Toto extends Object implements Tutu {
    private int i;
    public Toto(int j) {
        super();
        this.i = j;
    }
    public static void main(String[] toto) {
        Tete t = new Toto();
        System.out.println(t.m1());
        System.out.println(t.m2());
    }
}

class Tete extends Toto {
    private int i;
    public Tete() {
        super();
        this.i = 2;
    }
    public int m1() {
        return this.i;
    }
    public int m2() {
        return super.i;
    }
}
```