



Khóa học

Thi cuối kỳ 2022-1 (IT3100,
IT3103, IT3100Q)

hi



Nội dung thi

Nội dung thi

#d7e70c

1 điểm (có tích lũy, không hiển thị kết quả)

Kết quả in ra màn hình khi thực thi đoạn code sau là gì

```
System.out.print("Begin ");  
try {  
    int i = 5/0;  
    System.out.print("Try ");  
} catch (Exception ex) {  
    System.out.print("Catch ");  
} finally {  
    System.out.print("Finally ");  
}  
System.out.print("End ");
```

- ☐ Begin End
- ☐ Begin Try Catch Finally End
- ☒ Begin Catch Finally End
- ☐ Begin Catch End
- ☐ Begin Try Catch End

Gửi

#8bab79

1 điểm (có tích lũy, không hiển thị kết quả)

Đoạn mã sau nằm trong hàm `main` . Nó in ra gì ?

```
int x = 7;  
int mssv = 10;  
final int z = x + mssv;  
z -= 1;  
System.out.println("Value of z : " + z );
```

- ☒ Báo lỗi
- ☐ Value of z : 16
- ☐ Value of z : 17

Gửi

#500658

1 điểm (có tích lũy, không hiển thị kết quả)

Những ngôn ngữ lập trình nào hỗ trợ đa thừa kế

- ☐ Java
- ☐ C#
- ☒ C++
- ☐ Không phải ngôn ngữ nào trong 3 ngôn ngữ Java, C#, C++ hỗ trợ đa thừa kế

Gửi

#6a0513

1 điểm (có tích lũy, không hiển thị kết quả)

Chọn những phát biểu đúng về xử lý ngoại lệ trong Java

- ☒ Xử lý ngoại lệ giúp chương trình đáng tin cậy hơn, tránh kết thúc bất thường
- ☒ Mô hình xử lý ngoại lệ với try catch giúp tách biệt khối lệnh có thể xảy ra ngoại lệ và khối lệnh xử lý ngoại lệ
- ☐ Khi xảy ra ngoại lệ, đối tượng ngoại lệ được tạo ra và truyền đến khối catch
- ☒ Khi xảy ra ngoại lệ, ngoại lệ bắt buộc phải được xử lý ngay ở phương thức xảy ra ngoại lệ

Gửi

#9168ba

1 điểm (có tích lũy, không hiển thị kết quả)

Phát biểu nào đúng về khởi tạo và hủy bỏ đối tượng trong Java

- ☒ Trong Java không có phương thức hủy
- ☒ Java có phương thức có thể dùng thay thế phương thức hủy là phương thức finalize
- ☒ Một lớp luôn luôn có ít nhất 1 phương thức khởi tạo
- ☒ Mỗi đối tượng khi tồn tại và hoạt động được hệ điều hành cấp phát một vùng nhớ để lưu lại các giá trị của dữ liệu thành phần
- ☒ Các thuộc tính kiểu DL nguyên thủy (int, char, boolean, ...) của đối tượng phải được Lập trình viên khởi tạo trước khi sử dụng

Gửi

#fc23e9

1 điểm (có tích lũy, không hiển thị kết quả)

Cho lớp Person như sau

```
package p1;
public class Person {
    private int a1;
    int a2;
    protected int a3;
    public int a4;
}
```

Lớp Employee sau đây có lỗi ở những dòng nào

```
1. package p2;
2. import p1.Person;
3. public class Employee extends Person{
4.     public void print() {
5.         System.out.println(a1);
6.         System.out.println(a2);
7.         System.out.println(a3);
8.         System.out.println(a4);
9.     }
10. }
```

- ☒ 5
- ☒ 6
- ☐ 7
- ☐ 8
- ☐ Không bị lỗi ở dòng nào

Gửi

#76e584

1 điểm (có tích lũy, không hiển thị kết quả)

Các chỉ định truy cập nào có thể được dùng cho thành viên lớp trong Java

- ☒ public
- ☒ default
- ☒ protected
- ☒ private
- ☐ internal
- ☐ protected internal

Gửi

#b2a7b5

1 điểm (có tích lũy, không hiển thị kết quả)

Cho 4 phương thức sau

```
public void check1 (int n) throws Exception {
    if (n <= 0)
        throw new Exception("Number should be greater than 0");
}
public void check2 (int n) throws RuntimeException {
    if (n == 5)
        throw new RuntimeException("Number should not be 5");
}
public void check3 (int n) throws Exception, RuntimeException {
    check1(n);
    check2(n);
}
public void test() {
    int n = 4;
    check1(n);
    check2(n);
    check3(n);
}
```

Phương thức test bị lỗi ở những đâu

- ☒ Lỗi gọi phương thức check1
- ☐ Lỗi gọi phương thức check2
- ☒ Lỗi gọi phương thức check3
- ☐ Không bị lỗi gì

Gửi

#3bd76f

1 điểm (có tích lũy, không hiển thị kết quả)

Đoạn mã sau khi chạy in ra gì ?

```
class A {
    int i = 200;
}

class B extends A {
    int i = 100;
}

public class Test {
    public static void main(String[] ss) {
        B a = new B();
        System.out.println(a.i);
    }
}
```

- ☐ Báo lỗi biên dịch
- ☒ 100
- ☐ 200

Gửi

#11a90c

1 điểm (có tích lũy, không hiển thị kết quả)

Đâu là các ấn bản Java phát hành bởi Oracle

- 9 .
- ☐ Java EE
 - ☐ Java SE
 - ☐ Java ME
 - ☐ Java FX
 - ☐ Java Virtual Machine

Gửi

#74135d

1 điểm (có tích lũy, không hiển thị kết quả)

Kết quả in ra màn hình khi thực thi đoạn code sau là gì

```
System.out.print("Begin ");
try {
    int i = 5/0;
    System.out.print("Try ");
} finally {
    System.out.print("Finally ");
}
System.out.print("End ");
```

- ☐ Begin End
- ☐ Begin Finally End
- ☐ Begin Try Finally End
- ☐ Begin Try End
- ☒ Begin Finally + thông báo kết thúc chương trình

Gửi

#005e56

1 điểm (có tích lũy, không hiển thị kết quả)

Đoạn mã sau bị lỗi biên dịch ở những dòng nào ?

```
1.      class A {
2.          public int check(double a) {
3.              return (int) a;
4.          }
5.          public double check (double a) {
6.              return a;
7.          }
8.      }
9.      class B {
10.         private int check (double a, int b) {
11.             return (int) a + b;
12.         }
13.         private double check (double a, int b) {
14.             return a + b;
15.         }
16.     }
```

- ☐ Dòng 2

- ☐ Dòng 3
- ☒ Dòng 5
- ☐ Dòng 6
- ☐ Dòng 10
- ☐ Dòng 11
- ☒ Dòng 13
- ☐ Dòng 14

Gửi

#8f9b5f

1 điểm (có tích lũy, không hiển thị kết quả)

Chọn những phát biểu đúng về kỹ thuật ghi đè (overriding)

- ☒ Phương thức ở lớp con hoàn toàn giống về chữ ký với phương thức kế thừa ở lớp cha
- ☒ Phương thức lớp con KHÔNG bắt buộc có cùng kiểu trả về với phương thức kế thừa trong lớp cha
- ☒ Chỉ định truy cập ở lớp con KHÔNG giới hạn chặt hơn chỉ định truy cập phương thức kế thừa trong lớp cha
- ☒ Phương thức lớp con KHÔNG tung ra kiểu ngoại lệ mới so với phương thức kế thừa trong lớp cha

Gửi

#06e281

1 điểm (có tích lũy, không hiển thị kết quả)

Kết quả thực thi chương trình sau là gì

```
public class Test {  
    public static void p(String s) {  
        System.out.print(s + " ");  
    }  
    public static void f(int i) { p("int"); }  
    public static void f(short h) { p("short"); }  
    public static void f(char c) { p("char"); }  
    public static void f(long l) { p("long"); }  
    public static void f(double l) { p("double"); }  
    public static void f(float f) { p("float"); }  
    public static void main(String[] args) {  
        f(5);  
        f('c');  
        f(5.5);  
        f(7L);  
    }  
}
```

- ☒ int char double long
- ☐ int int float long
- ☐ short char float long
- ☐ Lỗi biên dịch ở 1 trong các lời gọi hàm f (trong hàm main)

Gửi

#9acad7

1 điểm (có tích lũy, không hiển thị kết quả)

Kết quả khi thực thi chương trình sau là gì

```
class Test {  
    public static void increase1(int[] arr) {  
        for (int a: arr)  
            a += 1;  
    }  
    public static void increase2(int[] arr) {  
        for (int i=0; i<arr.length; ++i)  
            arr[i] ++;  
    }  
    public static void main(String[] args) {  
        int arr[] = {5, 6, 7};  
        increase1(arr);  
        increase2(arr);  
        for (int a: arr)  
            System.out.print(a + " ");  
    }  
}
```

☐ 5 6 7

☒ 6 7 8

☐ 7 8 9

☐ 4 5 6

Gửi

#60f96f

1 điểm (có tích lũy, không hiển thị kết quả)

Chọn phát biểu đúng về lập trình tổng quát trong Java

- ☒ Có thể lập trình tổng quát cho lớp
- ☒ Có thể lập trình tổng quát cho phương thức
- ☒ Một lớp có thể khai báo số lượng kiểu dữ liệu tổng quát tùy ý
- ☒ Một phương thức chỉ có thể khai báo tối đa 1 kiểu dữ liệu tổng quát
- ☒ Không thể khai báo ràng buộc kiểu dữ liệu tổng quát cho lớp (VD lớp sẽ làm việc với kiểu dữ liệu T bất kỳ, sao cho T là Person hoặc lớp con của lớp Person)

Gửi

#d9cb95

1 điểm (có tích lũy, không hiển thị kết quả)

Cho 3 lớp: lớp Person, lớp Employee kế thừa lớp Person, và lớp Manager kế thừa lớp Employee. Chọn TẤT CẢ những phát biểu đúng về đoạn code sau

```
1.    Person p1 = new Manager();  
2.    Employee e = (Employee) p1;  
3.    Person p2 = new Employee();  
4.    Manager m = (Manager) p2;
```

☒ Dòng 1 là upcasting

- ☐ Dòng 2 là upcasting
- ☒ Dòng 2 là downcasting
- ☒ Dòng 3 là upcasting
- ☐ Dòng 3 là downcasting
- ☐ Dòng 4 là upcasting
- ☒ Dòng 4 là downcasting
- ☐ Không dòng nào bị lỗi biên dịch
- ☐ Không dòng nào bị lỗi thực thi

Gửi

#009f5e

1 điểm (có tích lũy, không hiển thị kết quả)

Đoạn mã sau in ra gì ?

```
public class A {
    String s;
    A a;

    public A(String s){
        this.s = s;
    }

    public static void main(String[] args) {
        A a1 = new A("first");
        A a2 = new A("second");
        a1.a = a2;
        a2.a = a1;
        System.out.print(a1.a.s);
        System.out.print(a2.a.s);
    }
}
```

- ☐ Báo lỗi
- ☒ secondfirst
- ☐ nullnull
- ☐ firstsecond

Gửi

#6f613c

1 điểm (có tích lũy, không hiển thị kết quả)

Sắp xếp các kiểu dữ liệu char, int, byte, short, long, float, double theo chiều nói rộng dần dữ liệu

- ☐ byte short char long float double
- ☐ byte char short long float double
- ☒ byte char short float long double
- ☐ byte short char float long double
- ☐ Không phải là đáp án nào trong các đáp án còn lại

Gửi

☐ Câu trả lời đã được gửi.

#1b5c7f

1 điểm (có tích lũy, không hiển thị kết quả)

Đoạn mã sau khi chạy in ra gì ?

```
interface P
{
    String p = "PPPP";
    String methodP();
}

interface Q extends P
{
    String q = "QQQQ";
    String methodQ();
}

class R implements P, Q
{
    public String methodP()
    {
        return q+p;
    }

    public String methodQ()
    {
        return p+q;
    }
}

public class Demo {
    public static void main(String[] ss) {
        R r = new R();
        System.out.print(r.methodP());
        System.out.print(r.methodQ());
    }
}
```

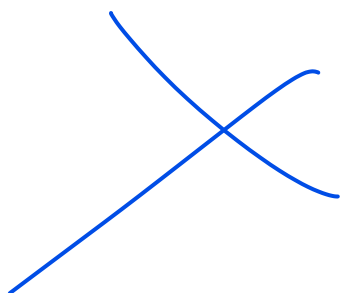
- ☒ QQQQPPPPPPPPQQQQ
- ☐ PPPPQQQQPPPPQQQQ
- ☐ Báo lỗi biên dịch
- ☐ QQQQPPPPQQQQPPPP

Gửi

#f208c1

1 điểm (có tích lũy, không hiển thị kết quả)

Đoạn mã sau khi chạy in ra gì ?



```
public class Test {
    public static void main(String[] ss) {
        String[] s = {"abc", "123", null};
        for (int i = 0; i < s.length; i++){
            try {
                int a = s[i].length();
                try {
                    a = Integer.parseInt(s[i]);
                } catch (NumberFormatException ex) {
                    System.out.print("NumberFormatException ");
                }
            } catch (NullPointerException ex) {
                System.out.print("NullPointerException ");
            }
        }
    }
}
```

- ☐ NullPointerException NumberFormatException
- ☐ abc 123 null
- ☒ NumberFormatException NullPointerException
- ☐ abc 123

Gửi

#2ed507

1 điểm (có tích lũy, không hiển thị kết quả)

Kết quả thực thi đoạn chương trình sau là gì

```
public class Test {
    static int index = 0;

    void increase() {
        index++;
    }

    public static void main(String[] args) {
        increase();
        System.out.println(index);
    }
}
```

- ☐ In ra 1
- ☐ In ra 0
- ☒ Lỗi biên dịch ở lời gọi hàm increase trong hàm main
- ☐ Lỗi biên dịch ở lệnh in ra giá trị index trong hàm main

Gửi

#970e32

1 điểm (có tích lũy, không hiển thị kết quả)

Đoạn mã sau khi chạy in ra gì?

```
public class Test{
    public static void tryPrimitives(int i, double f, char c, boolean test)
    {
        i += 10;
        c = 'z';
        if(test)
            test = false;
        else
            test = true;
    }

    public static void main(String[] args)
    {
        int ii = 1;
        double ff = 1.0;
        char cc = 'a';
        boolean bb = false;
        tryPrimitives(ii, ff, cc, bb);
        System.out.println("ii = " + ii + ", ff = " + ff +
            ", cc = " + cc + ", bb = " + bb);
    }
}
```

Handwritten notes:
- A circle around the parameter `int i` in the `tryPrimitives` method signature, with an arrow pointing to the word "Integer" written below it.
- A vertical line next to the `tryPrimitives` method call in the `main` method.

- ☐ ii = 11, ff = 0.0, cc = z, bb =true
- ☒ ii = 1, ff = 1.0, cc = a, bb = false
- ☐ ii = 1, ff = 0, cc = z, bb =true
- ☐ ii = 11, ff = 1.0, cc = a, bb = false

Gửi

#062b84

1 điểm (có tích lũy, không hiển thị kết quả)

Cho lớp Person như sau

```
class Person {
    protected String name;
    private String print1() {
        return name;
    }
    protected String print2() {
        return name.toUpperCase();
    }
    protected String print3() {
        return name.toLowerCase();
    }
}
```

Chọn những phát biểu đúng về lớp Employee sau đây:

```
class Employee extends Person {
    private void print1() {
        System.out.println(name);
    }
    protected void print2() {
        System.out.println("Name: " + name.toUpperCase());
    }
    String print3() {
        return "Name: " + name.toLowerCase();
    }
}
```

- ☐ Lỗi biên dịch ở phương thức print1
- ☒ Lỗi biên dịch ở phương thức print2
- ☒ Lỗi biên dịch ở phương thức print3
- ☐ Không bị lỗi gì cả

Gửi

#8dcf75

1 điểm (có tích lũy, không hiển thị kết quả)

Chọn phát biểu đúng về xử lý sự kiện trong JavaFX

- ☒ JavaFX hỗ trợ xử lý sự kiện chuột, bàn phím với 2 lớp MouseEvent và KeyEvent
- ☒ Để xử lý sự kiện rê chuột (drag), sử dụng lớp DragEvent
- ☒ JavaFX cung cấp các handlers và các filters để xử lý sự kiện
- ☒ Khi xử lý sự kiện, các handler được gọi trước, các filter được gọi sau
- ☒ Tất cả các handlers đều thực thi giao diện javafx.event.EventHandler. Trong khi đó, các filters cần thực thi một giao diện khác.
- ☒ Một node có thể đăng ký nhiều handler/filter

Gửi

#1b053a

1 điểm (có tích lũy, không hiển thị kết quả)

Chọn những phát biểu đúng về 2 phương thức sau

```
public void print1(ArrayList<Employee> list) {
}
public void print2(ArrayList<? extends Employee> list) {
}
```

- ☐ 2 phương thức là hoàn toàn tương đương nhau
- ☐ Trong phương thức print1 không thể thêm mới phần tử vào list. Và trong phương thức print2 có thể thêm mới phần tử vào list
- ☐ Trong phương thức print2 không thể thêm mới phần tử vào list. Và trong phương thức print1 có thể thêm mới phần tử vào list
- ☒ Để gọi phương thức print2, danh sách truyền vào chỉ được chứa các đối tượng là lớp con của Employee (không tính Employee)

Gửi

#ef13ed

1 điểm (có tích lũy, không hiển thị kết quả)

Kết quả khi thực thi lớp Test là gì

```
class Person {
    private int age;
    public Person(int age) {
        this.age = age;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
}
class Test {
    public static void increaseAge(Person p) {
        p.setAge(p.getAge() + 1);
    }
    public static void swap(Person p1, Person p2) {
        Person tmp = p1;
        p1 = p2;
        p2 = tmp;
    }
    public static void main(String[] args) {
        Person p1 = new Person(15);
        Person p2 = new Person(20);
        increaseAge(p1);
        swap(p1, p2);
        System.out.print(p1.getAge() + " ");
        System.out.print(p2.getAge() + " ");
    }
}
```

p: lura a >c

^ ^ tach omm!!!

><

- ☐

15 20
- ☒

16 20
- ☐

20 15
- ☐

20 16

Gửi

#bce44e

1 điểm (có tích lũy, không hiển thị kết quả)

Kết quả in ra màn hình khi thực thi lớp Employee là gì

Suy!

```

class Person {
    protected int age;
    protected String name;

    public Person() {
        System.out.print(age + " " + name + " ");
    }
    public Person(int age, String name) {
        this.age = age;
        this.name = name;
    }
}
class Employee extends Person{
    public Employee() {
    }
    public Employee(int age, String name) {
        super(age, name);
    }
    public static void main(String[] args) {
        Employee e1 = new Employee();
        Employee e2 = new Employee(16, "Huy");
        System.out.print(e2.age + " " + e2.name);
    }
}

```

- ☐ 16 Huy
- ☒ 0 null 16 Huy
- ☐ Lỗi biên dịch ở hàm main
- ☒ 16 Huy 16 Huy

Gửi

#300730

1 điểm (có tích lũy, không hiển thị kết quả)

Cho lớp Person như sau

```

package p1;
public class Person {
    private int a1;
    int a2;
    protected int a3;
    public int a4;
}

```

Lớp Employee sau đây có lỗi ở những dòng nào

```

1.    package p2;
2.    import p1.Person;
3.    public class Employee extends Person{
4.        public void print(Person p) {
5.            System.out.println(p.a1);
6.            System.out.println(p.a2);
7.            System.out.println(p.a3);
8.            System.out.println(p.a4);
9.        }
10.   }

```

- ☒ 5
- ☒ 6

- ☐ 7
- ☐ 8
- ☐ Không bị lỗi ở dòng nào

Gửi

#b89bfb

1 điểm (có tích lũy, không hiển thị kết quả)

Đoạn mã sau khi chạy in ra gì ?

```
class A {  
    String myMethod(String s) {  
        return s+s;  
    }  
}  
  
class B extends A {  
    String myMethod(String s, double d) {  
        return myMethod(s+d);  
    }  
}  
  
class C extends B {  
    String myMethod(String s, double d, int i) {  
        return myMethod(s, d+i);  
    }  
}  
  
public class Test {  
    public static void main(String[] ss) {  
        C x = new C();  
        System.out.println(x.myMethod("CHECK", 12.34, 56));  
    }  
}
```

A
↓
B
↓
C

- ☒ CHECK68.34CHECK68.34
- ☐ CHECK12.34+56CHECK12.34+56
- ☐ CHECK12.34+56
- ☐ CHECK68.34

Gửi

#0370d2

1 điểm (có tích lũy, không hiển thị kết quả)

Đoạn mã sau khi chạy in ra gì ?

```
class A {  
  
}  
  
class B extends A {  
  
}  
  
class C extends B {  
  
}  
  
public class Test {  
    static void method(A a) {  
        System.out.println("Method A");  
    }  
  
    static void method(B b) {  
        System.out.println("Method B");  
    }  
  
    static void method(Object obj) {  
        System.out.println("Method C");  
    }  
    public static void main(String[] ss) {  
        C c = new C();  
        method(c);  
    }  
}
```

- ☐ Method C
- ☒ Method B
- ☐ Method A

Gửi

#90e310

1 điểm (có tích lũy, không hiển thị kết quả)

Cho 2 interface như sau

```
interface Printable {  
    public void print();  
}  
interface Stringable {  
    String stringify();  
}
```

Lớp Person sau bị lỗi ở những dòng nào


```
1.      class Person implements Printable, Stringable {
2.          protected String name;
3.          public String stringify() {
4.              return name;
5.          }
6.          public void print() {
7.              System.out.println(name);
8.          }
9.          public static void main(String[] args) {
10.              Printable p1 = new Printable();
11.              Printable p2 = new Person();
12.              String name = p2.stringify();
13.          }
14.      }
```

☐ Dòng 1

☐ Dòng 6

☒ Dòng 10

☐ Dòng 11

☒ Dòng 12

Gửi