

BÀI TẬP 3

Trả lời câu hỏi

1. Không có output
2. In ra Test Exception
3. Xảy ra runtime exception
4. Lỗi biên dịch

```
public class Test extends Exception{
    private String message;

    public Test(String message){
        this.message = message;
    }

    public static void main(String[] args){
        int a = 5, b = 3;
        try{
            if( a % b > 0 )
                throw new Test();
        }catch(Exception e){
            System.out.println("Test Exception");
        }
    }
}
```

Trả lời câu hỏi

1. Exception 2
2. Exception 2 Exception 1
3. 10
4. Lỗi biên dịch

```
public class Test{
    public static void main(String[] args){
        try{
            Test t = new Test();
            int d = t.getNum(2, 5);
            System.out.println(d);
        }catch(Exception e){
            System.out.print("Exception 1 ");
        }
    }

    public int getNum(int a, int b){
        int c = 0;
        try{
            c = a * b;
            if(c > 10)
                throw new String("Cannot be more than 10 ");
        }catch(Exception e){
            System.out.print("Exception 2 ");
        }
        return c;
    }
}
```

Trả lời câu hỏi

1. Test
2. Exception
3. Null
4. Lỗi biên dịch

```
public class Test{
    String className;

    public static void main(String[] args){
        try{
            Test t = new Test();
            if(t.className.equals("Test"))
                System.out.print("Test ");
            else
                System.out.print("Other ");
        }catch(Exception e){
            System.out.print("Exception ");
        }catch(NullPointerException ne){
            System.out.print("Null ");
        }
    }
}
```

Trả lời câu hỏi

1. Jack Name cannot be John
2. Jack NamelsJohnException
3. NamelsJohnException
4. Lỗi biên dịch

```
public class Test{
    public static void main(String[] args){
        try{
            String[] names = {"Jack", "John", "Jill"};
            printNames(names);
        }catch(NameIsJohnException e){
            System.out.print("NameIsJohnException ");
        }
    }

    private static void printNames(String[] names) throws NameIsJohnException{
        for(String name : names){
            if(name.equals("John"))
                throw new NameIsJohnException("Name cannot be John ");
            System.out.print(name + " ");
        }
    }
}

class NameIsJohnException{
    String message;

    public NameIsJohnException(){
    }

    public NameIsJohnException(String message){
        this.message = message;
    }

    public String getMessage(){
        return this.message;
    }
}
```

Trả lời câu hỏi

1. Không có output
2. I have done nothing
3. Exception1
4. Lỗi biên dịch

```
import java.io.IOException;

public class Test{
    public static void main(String[] args){
        try{
            Test t = new Test();
            t.doNothing();
            System.out.println("I have done nothing");
        }catch(IOException e){
            System.out.println("Exception1");
        }
    }

    private void doNothing(){
        for(int i = 0 ; i < 10; i++){
        }
    }
}
```

Trả lời câu hỏi

1. Không có output
2. 5
3. Exception – Important data is invalid
4. Lỗi biên dịch

```
public class Test extends Exception{
    public Test(){
    }
    public Test(String str){
        super(str);
    }

    int importantData = 5;
    public static void main(String[] args){
        Test t = new Test();
        t.importantMethod();
    }

    private void importantMethod(){
        if( importantData > 5)
            throw new Test("Important data is invalid");
        else
            System.out.println(importantData);
    }
}
```

Trả lời câu hỏi

1. Exception caught
2. Method 1 ends
3. Main Completed
4. Lỗi biên dịch

```
public class Test {  
    public static void main(String[] args){  
        try{  
            AnotherClass obj = new AnotherClass();  
            obj.method1();  
            System.out.println("Main Completed");  
        }catch (Exception e){}  
    }  
}  
  
class AnotherClass{  
    public void method2(){  
        throw new ArrayIndexOutOfBoundsException();  
    }  
  
    public void method1(){  
        try{  
            method2();  
        }catch (NullPointerException ae){  
            System.out.println("Exception caught");  
        }finally{  
            System.out.println("Method 1 ends");  
        }  
    }  
}
```


Trả lời câu hỏi

1. Không có output
2. Exception
3. Exception Finally
4. Lỗi biên dịch

```
public class Test {  
    public static void main(String[] args){  
        try{  
            Test t = new Test();  
            t.method1();  
        }catch (Exception e){  
            System.out.print("Exception ");  
        }finally{  
            System.out.print("Finally ");  
        }  
    }  
  
    public void method1(){  
        throw new RuntimeException();  
    }  
}
```

Trả lời câu hỏi

1. Exception Finally
2. Ném ra AssertionError khi chạy
3. Ném ra AssertionError và in ra Finally
4. Lỗi biên dịch

```
public class Test {  
    public static void main(String[] args){  
        try{  
            Test t = new Test();  
            t.method1();  
        }catch (Exception e){  
            System.out.print("Exception ");  
        }finally{  
            System.out.print("Finally ");  
        }  
    }  
  
    public void method1(){  
        throw new AssertionError();  
    }  
}
```

Trả lời câu hỏi

1. 0
2. 0 Finally
3. Finally 0
4. Lỗi biên dịch

```
public class Test {  
    private int i;  
  
    public static void main(String[] args){  
        Test t = new Test();  
        System.out.println(t.method1());  
    }  
  
    public int method1(){  
        try{  
            if(i == 0)  
                return 0;  
            else  
                return 10/i;  
        }catch (Exception re){  
            System.out.print("Exception ");  
        }finally{  
            System.out.print("Finally ");  
        }  
        return 0;  
    }  
}
```

Trả lời câu hỏi

1. 3
2. 4
3. 5
4. 6

```
public class Test {  
    public int i;  
  
    public static void main(String[] args){  
        Test t = new Test();  
        try{  
            t.method1();  
        }catch(Exception e){  
            t.here();  
        }finally{  
            t.here();  
        }  
        System.out.println(t.i);  
    }  
  
    public void method1(){  
        try{  
            method2();  
            here();  
        }catch(Exception e){  
            here();  
        }finally{  
            here();  
        }  
    }  
  
    public void method2(){  
        try{  
            throw new Exception();  
        }catch(Exception e){  
            here();  
        }finally{  
            here();  
        }  
    }  
  
    public void here(){  
        i++;  
    }  
}
```

Thảo luận

Một phần mềm bán hàng cho phép khách hàng chọn mã mặt hàng và số lượng. Người dùng có thể gặp phải các lỗi sau:

1. Mã mặt hàng không phải kiểu số, quá thấp (dưới 0) hoặc quá cao (trên 9999).
2. Số lượng không phải kiểu số, quá thấp (dưới 1) hoặc quá cao (trên 99).
3. Mã mặt hàng không có trong danh sách.

Viết chương trình cho phép người dùng chọn mã và số lượng và đề xuất cách xử lý các lỗi trên.