

Chapter 11 异常

Key Point :

- 异常的概念和分类
- 异常的产生和传递
- 异常的处理
- 自定义异常

问题：

1. 填空

Java 中所有的错误都继承自_____类；在该类的子类中，_____类表示严重的底层错误，对于这类错误一般处理的方式是_____；_____类表示例外、异常。

2. 查询 API，填空

- 异常类 `java.rmi.AlreadyBoundException`，从分类上说，该类属于_____（已检查|运行时）异常，从处理方式上说，对这种异常_____处理。
- 异常类 `java.util.regex.PatternSyntaxException`，从分类上说，该类属于_____（已检查|运行时）异常，从处理方式上说，对这种异常_____处理。

3. （异常的产生）把下面代码补充完整

```
public class TestThrow{  
  
    public static void main(String args[]){  
  
        throwException(10);  
    }  
}
```

```

    }

    public static void throwException(int n){

        if (n == 0){

            //抛出一个 NullPointerException

        }else{

            //抛出一个 ClassCastException

            //并设定详细信息为 “类型转换出错”

        }

    }

}

```

4. （ try-catch-finally ） 有如下代码：

```

import java.io.*;

import java.sql.*;

class TestException{

    public static void main(String args[]){

        System.out.println( "main 1" );

        int n;

        //读入 n

        ma(n);

        System.out.println( "main2" );

    }

    public static void ma(int n){

        try{

```

```

        System.out.println( "ma1" );

        mb(n);

        System.out.println( "ma2" );
    }catch(EOFException e){

        System.out.println( "Catch EOFException" );
    }catch(IOException e){

        System.out.println( "Catch IOException" );
    }catch(SQLException e){

        System.out.println( "Catch SQLException" );
    }catch(Exception e){

        System.out.println( "Catch Exception" );
    }finally{

        System.out.println( "In finally" );
    }
}

```

```

public static void mb(int n) throws Exception{

    System.out.println( "mb1" );

    if (n == 1) throw new EOFException();

    if (n == 2) throw new FileNotFoundException();

    if (n == 3) throw new SQLException();

    if (n == 4) throw new NullPointerException();

    System.out.println( "mb2" );

}

```

```

}

```

问：当读入的 n 分别为 1, 2, 3, 4, 5 时，输出的结果分别是什么？

5. （自定义异常）创建两个自定义异常类 MyException1 和 MyException2。

要求：

- I. MyException1 为已检查异常，MyException2 为运行时异常。
- II. 这两个异常均具有两个构造函数，一个无参，另一个带字符串参数，参数表示产生异常的信息。

6. （自定义异常）在上一题的基础上，把下面代码补充完整。

```
public class TestMyException{

    public static void main(String args[]){

        int n;

        //读入 n

        try{

            m(n);

        }catch(MyException1 ex1){

            //输出 ex1 详细的方法调用栈信息

        }catch(MyException2 ex2){

            //输出 ex2 的详细信息

            //并把 ex2 重新抛出

        }

    }

    public static void m(int n)_____ { //声明抛出 MyException1

        if (n == 1) {

            //抛出 MyException1
```

```

        //并设定其详细信息为 "n == 1"

    }else {

        //抛出 MyException2

        //并设定其详细信息为 "n == 2"

    }

}

}

```

7. (try-catch) 代码改错。

```

class MyException{}

class TestException{

    public static void main(String args[]){

        ma();

    }

    public static int ma(){

        try{

            m();

            return 100;

        }catch(Exception e){

            System.out.println( "Exception" );

        }

        catch(ArithmeticException e){

            System.out.println( "ArithmeticException" );

        }

    }

}

```

```
    }  
  
    public static void m(){  
        throw new MyException();  
    }  
}
```

8. （方法覆盖）有如下代码

```
import java.io.IOException;  
  
class Super{  
    public void ma() throws IOException{}  
}  
  
interface IA{  
    void mb();  
}  
  
public class MySub extends Super implements IA{  
    public void ma() //1____{}  
    public void mb() //2____{}  
}
```

问：在//1 处，填入以下____代码可以编译通过，在//2 处，填入____代码可以编译通过。

- A. throws java.io.IOException
- B. throws java.io.FileNotFoundException, java.io.EOFException
- C. throws java.sql.SQLException
- D. 不能抛出任何异常。

9. (try-catch , 局部变量) 有如下代码

```
public class TestTryCatch{

    public static void main(String args[]){

        System.out.println( ma() );

    }

    public static int ma(){

        int n;

        try{

            n = 10/0;

        }catch(Exception e){

        }

        return n;

    }

}
```

选择正确答案：

- A. 编译不通过
- B. 编译通过，输出-1
- C. 编译通过，输出 0

10. (try-catch-finally) 有如下代码

```
public class TestFinally{

    public static void main(String args[]){

        System.out.println ( ma() );

    }

}
```

```

public static int ma(){

    int b;

    //读入 b

    try{

        int n = 100;

        return n/b;

    }catch(Exception e){

        return 10;

    }finally{

        return 100;

    }

}
}

```

在 ma 中，当读入的 b 为 100 时，输出结果为____，当读入的 b 为 0 时，输出结果为_____。

11. (try-finally) 写出下面代码运行的结果

```

public class TestTryFinally{

    public static void main(String args[]){

        try{

            ma();

        }catch(Exception ex1){

        }

    }

    public static void ma() throws Exception {

```



```

int n = 10;

int b;

//读入一个整数 b

try{

    System.out.println( "ma1" );

    int result = n / b;

    System.out.println( "ma2 " + result);

}finally{

    System.out.println( "In Finally" );

}

}

}

```

在 ma 中，读入整数 b，如果读入的值为 10，则输出： ____ 。如果读入的值为 0，则输出： ____ 。

12. （方法覆盖）

```

import java.io.*;

class MySuper{

    public void m() throws IOException{}

}

class MySub extends MySuper{

    public void m() throws EOFException{}

}

class MySub2 extends MySub {

    public void m() throws FileNotFoundException{}

}

```

```
}
```

以上代码是否能编译通过？如果不能，应该如何修改？

13. （异常的捕获和抛出）有以下代码

```
public class TestException{  
    public static void main(String args[]){  
        try{  
            System.out.println( "main1" );  
            ma();  
            System.out.println( "main2" );  
        }catch(Exception e){  
            System.out.println( "In Catch" );  
        }  
    }  
  
    public static void ma(){  
        System.out.println( "ma1" );  
        throw new NullPointerException();  
        System.out.println( "ma2" );  
    }  
}
```

选择正确答案：

- A. 编译出错
- B. 编译正常，输出 main1 ma1 In Catch
- C. 编译正常，运行时出错

14. (异常的捕获和抛出) 有如下代码

```
import java.io.*;

import java.sql.*;

class TestException{

    public static void main(String args[]){

        try{

            ma();

        }

        /*1*/

        catch(Exception e){

        }

    }

    public static void ma() throws IOException{ }

}
```

下面哪些代码放在/*1*/处可以编译通过？

- A. catch(NullPointerException npe){}
- B. catch(IOException ioe){}
- C. catch(SQLException sqle){}