

Day12

Key Point

- 包装类
 - 自动封箱拆箱
- 数学类
 - Math
 - BigDecimal
- 日期类
 - Date
 - Calendar
- 异常
 - Error
 - Exception
 - 编译时异常(已检查异常)
 - 运行时异常(未检查异常)

一、编程题

1. 对于字符串String s = "2016-11-11 00:11:11" , 下列能够将其正确转化为日期对象2016年11月11日0时11分11秒的选项是:
 - A. Date d = new Date(s);
 - B. Date date = new SimpleDateFormat("yyyy/mm/dd hh:MM:ss").parse();
 - C. Date date = new SimpleDateFormat("yyyy-mm-dd hh:MM:ss").parse();
 - D. Date date = new SimpleDateFormat("YYYY-mm-dd hh:MM:ss").parse();
 - E. Date date = new SimpleDateFormat("yyyy-mm-dd HH:MM:ss").parse();
2. 下列各项能够正确编译运行的是:
 - A. Integer i = new Integer("3");
 - B. Float f = 3.5;
 - C. Integer i = new Integer("0x45");
 - D. Boolean b = new Boolean(null);
 - E. Boolean b = new Boolean("aaa");
3. 对于代码:

```
public class Test {  
    public static void main(String[] args) {  
        System.out.println(m());  
    }  
    public static Person m() {  
        Person p = new Person();  
    }  
}
```

```

    try {
        p.setName("Amy");
        p.setAge(15);
        return p;
    } catch (Exception e) {
        p.setName("Sam");
        p.setAge(12);
        return p;
    } finally {
        p.setName("Tom");
        p.setAge(13);
    }
}
}

class Person {
    private String name;
    private int age;
    public String getName() {
        return name;
    }
    public void setName(String name) {
        this.name = name;
    }
    public int getAge() {
        return age;
    }
    public void setAge(int age) {
        this.age = age;
    }
    @Override
    public String toString() {
        return name + " " + age ;
    }
}
}

```

的运行结果是：

A. Amy 15 B. Sam 12 C. Tom 13 D. 编译错误 E. 运行错误

4. 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_____{  
}  
}
```

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

5. 有如下代码

```
import java.io.*;  
import java.sql.*;  
public class TestTryCatch{  
public static void main(String args[]){  
    try{  
        ma(10);  
        System.out.println("No Exception");  
    }  
    catch EOFException ex1){  
        System.out.println("ex1");  
    }  
    catch IOException ex2) {  
        System.out.println("ex2");  
    }  
    catch(SQLException ex3) {  
        System.out.println("ex3");  
    }  
}  
public static void ma(int n) throws Exception{  
    if (n == 1){  
        throw new IOException();  
    }else if (n == 2){  
        throw new EOFException();  
    }else if (n == 3) {  
        throw new SQLException();  
    }  
}  
}
```

选择正确答案：

- A. 编译不通过
- B. 编译通过，输出 No Exception
- C. 编译通过，输出 ex1
- D. 编译通过，输出 ex2
- E. 编译通过，输出 ex3

6. 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

7. 有如下代码

```

public class TestTryAndTry {
public static void main(String args[]){
    System.out.println(ma());
}
public static int ma(){
    try{
        return 100;
    }finally{
        try{
            return 200;
        }finally{
            return 500;
        }
        return 1000;
    }
}
}

```

选择正确答案:

- A. 编译错误
- B. 输出 100
- C. 输出 200
- D. 输出 500
- E. 输出 1000

二、简答题

1. 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. 异常类 `java.rmi.AlreadyBoundException`，从分类上说，该类属于_____（已检查| 未检查）异常，从处理方式上说，对这种异常_____；异常类 `java.util.regex.PatternSyntaxException`，从分类上说，该类属于_____（已检查|未检查）异常，从处理方式上说，对这种异常_____。
3. 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？

4. 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();  
}  
}
```

5. 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____。

6. 有以下代码：

```
import java.io.*;
import java.sql.*;
public class TestMyException{
public static void main(String args[]){
    try{
        System.out.println("main1");
        ma();
        System.out.println("main2");
    }catch(Exception e){
        System.out.println("Catch Exception in main");
        System.out.println(e.getMessage());
    }
}
public static void ma() throws IOException{
    try{
        System.out.println("ma1");
        mb();
        System.out.println("ma2");
    }catch(SQLException e){System.out.println("Catch SQLException in ma");
        throw new IOException(e.getMessage());
    }catch(Exception e){
        System.out.println("Catch Exception in ma");
        System.out.println(e.getMessage());
    }
}
public static void mb() throws SQLException{
    throw new SQLException("sql exception in mb");
}
}
```

问：该程序输出结果是什么？

三、编程题

- 从命令行上读入一个字符串，用两种不同的方法，把该字符串转换为一个int类型
方法一：把String 直接转换为int
方法二：把String 转换为Integer，再把Integer 转换为int 类型
- 创建两个自定义异常类 MyException1 和 MyException2。
要求：
 - MyException1 为已检查异常， MyException2 为未检查异常
 - 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详细信息。

