

一. 简答题

- 1
- 2
- 3

Java语言实习3 异常和异常处理

题量: 3 满分: 100

作答时间: 2023-09-18 10:15 至 2023-09-24 12:00

一. 简答题 (共3题)

1. (简答题)
2. 掌握异常处理方法（33分）。

(1) 捕获异常

运行下列程序，自定义类名，分析try-catch-finally语句的结果(5')。

```
public class _____
{
    public static void main(String args[ ])
    {
        fun(0);
        fun(1);
        fun(2);
        fun(3);
    }
    static void fun(int i)
    {
        System.out.println("调用方法:fun("+i+"");
        try
        {
            if(int i==0)
            {
                System.out.println("没有异常!");
            }
            else if(i==1)
            {
                int a=0;
                int b=10;
                b/=a;
            }
            else if(i==2)
            {
                int m[]=new int[5];
                m[5]=100;
            }
            else if(i==3)
            {
                String str="56k9";
                int n=Integer.parseInt(str);
            }
        }
        catch(ArithmeticException e)
        {
            System.out.println("捕捉异常："+e.getMessage());
        }
        catch(ArrayIndexOutOfBoundsException e)
        {
            System.out.println("捕捉异常："+e);
        }
        catch(NumberFormatException e)
        {
            System.out.println("捕捉异常："+e);
        }
        finally
        {
            System.out.println("处理完毕！\n");
        }
    }
}
```

程序分析

程序使用try-catch-finally来捕获并处理异常。其中有3个catch()方法，用来处理3种异常事件，请写出处理异常事件的类名和其作用，从输出结果分析代码如何触发异常。（2*4'）

使用以下catch()方法来代替程序中的3种catch()方法会出现什么结果？（3'）

```
catch(Exception e)
{
    System.out.println("捕获异常："+e);
}
```

- (2) 抛出异常
- 将（1）中的代码改写为throw抛出异常方法，类名throw_加姓名首字母，代码如下（5'）。

作业详情

```
static void fun1(int i) throws ArithmeticException
{
    System.out.println("调用方法:fun1("+i+")");
    if(i==0)
    {
        System.out.println("没有异常! \n");
    }
    else if(i==1)
    {
        int a=0;
        int b=10;
        b/=a;
        ArithmeticException e=new ArithmeticException();
        throw e;
    }
}
static void fun2() throws ArrayIndexOutOfBoundsException
{
    System.out.println("调用方法:fun2()");
    int m[]=new int[5];
    m[5]=100;
    ArrayIndexOutOfBoundsException e=new ArrayIndexOutOfBoundsException();
    throw e;
}
static void fun3() throws NumberFormatException
{
    System.out.println("调用方法:fun3()");
    String str="56k9";
    int n=Integer.parseInt(str);
    NumberFormatException e=new NumberFormatException();
    throws e;
}
public static void main(String args[])
{
    try
    {
        fun1(0);
        fun1(1);
    }
    catch(ArithmeticException e)
    {
        System.out.println("\t捕捉异常:"+e.getMessage()+"");
    }
    try
    { fun2(); }
    catch(ArrayIndexOutOfBoundsException e)
    {
        System.out.println("\t捕捉异常:"+e+"\n");
    }
    try
    { fun3(); }
    catch(NumberFormatException e)
    {
        System.out.println("\t捕捉异常:"+e);
    }
    finally
    {
        System.out.println("处理完毕! \n");
    }
}
```

程序分析

该程序和上一程序的区别在于程序对异常的处理不在产生异常的fun1()方法中进行,而在哪个方法进行?该程序中设定了哪些产生异常的方法?因此出现了3个try-catch-finally结构,哪些结构有finally语句?(3*4')

一. 简答题

1

2

3

作业详情

我的答案:

```
(1)
package error;

public class cxy {
    public static void main(String args[])
    {
        fun(0);
        fun(1);
        fun(2);
        fun(3);
    }
    static void fun(int i)
    {
        System.out.println("调用方法: fun("+i+")");
        try
        {
            if(i==0)
            {
                System.out.println("没有异常! ");
            }
            else if(i==1)
            {
                int a=0;
                int b=10;
                b/=a;
            }
            else if(i==2)
            {
                int m[]=new int[5];
                m[5]=100;
            }
            else if(i==3)
            {
                String str="56k9";
                int n=Integer.parseInt(str);
            }
        }
        catch(ArithmeticException e)
        {
            System.out.println("捕捉异常:"+e.getMessage());
        }
        catch(ArrayIndexOutOfBoundsException e)
        {
            System.out.println("捕捉异常:"+e);
        }
        catch(NumberFormatException e)
        {
            System.out.println("捕捉异常: "+e);
        }
        finally
        {
            System.out.println("处理完毕! \n");
        }
    }
}
```

实验结果

一. 简答题

1

2

3

terminated - C:\Program Files\Java\jdk-11.0.10\bin\java.exe (2023-11-11 16:07:07)

调用方法: fun(0)
没有异常!
处理完毕!

调用方法: fun(1)
捕捉异常: / by zero
处理完毕!

调用方法: fun(2)
捕捉异常: [java.lang.ArrayIndexOutOfBoundsException](#): 5
处理完毕!

调用方法: fun(3)
捕捉异常: [java.lang.NumberFormatException](#): For input string: "56k9"
处理完毕!

分析try-catch-finally语句的结果:

1. 运行下列程序, 自定义类名, 分析 try-catch-finally 语句的结果(5')。
- 运行与自定义图如上, 分析 try-catch-finally 语句的结果:
- try 语块用于捕获异常, catch 语块用于处理 try 捕获到的异常, 而 finally 语块则进行最终的执行, 无论是否出现异常。在 catch 中有三个异常类, 分别是
- ArithmeticException 异常类, 用于处理计算异常; 对应 fun(1) 方法由于 10 无法除以 0, 触发除法异常。因而结果为捕捉异常: /by zero (除以 0)。
- ArrayIndexOutOfBoundsException 异常类, 用于处理下标越界异常, 对应 fun(2) 方法中有五个元素的数组被定义为 m[5], 触发了下标越界异常。因而结果为捕捉异常: [java.lang.ArrayIndexOutOfBoundsException](#): 5
- [NumberFormatException](#) 异常类, 用于处理格式化数据异常, 对应 fun(3) 方法字符串 "56k9" 无法转换为 int 类型, 触发了数据异常。因而结果为捕捉异常: [java.lang.NumberFormatException](#): For input string: "56k9"
- 而 fun(0) 方法不触发除法异常和数据转换异常, 因而输出结果 "没有异常!"
- Finally 进行最终执行, 在每个方法中都输出了结果 "处理完毕!"

程序使用try-catch-finally来捕获并处理异常。其中有3个catch()方法, 用来处理3种异常事件, 请写出处理异常事件的类名和其作用, 从输出结果分析代码如何触发异常 (2*4')

2. 程序使用 try-catch-finally 来捕获并处理异常。其中有 3 个 catch() 方法, 用来处理 3 种异常事件, 请写出处理异常事件的类名和其作用, 从输出结果分析代码如何触发异常。(2*4')
- 第一种异常事件: ArithmeticException 异常类,
- 作用: 处理算术运算异常。由于 fun(1) 方法的结果为/by zero, 在代码出现了除以 0 的除法错误, 因而触发了算术运算异常, catch 捕捉到异常。
- 第二种异常事件: ArithmeticException 异常类,
- 作用: 处理下标越界异常。由于 fun(2) 方法的结果为 [java.lang.ArrayIndexOutOfBoundsException](#): 5, 说明这里的下标 5 超出了数组 m 的下标范围, 导致了触发了下标越界异常, 在 catch 中捕捉到了异常。
- 第三种异常事件: [ArrayIndexOutOfBoundsException](#) 异常类,
- 作用: 处理格式化数据异常。由于 fun(3) 的结果为 [java.lang.NumberFormatException](#): For input string: "56k9", 表示输入的字符串 "56k9" 是错误的, 无法转为所需类型 int, 因而触发了数据异常, 在 catch 中捕捉到了异常

使用以下catch()方法来代替程序中的3种catch()方法会出现什么结果? (3')

一. 简答题

- 1
- 2
- 3

作业详情

```
        System.out.println("捕获异常:"+e);
    }
}

调用方法: fun(0)
没有异常!
处理完毕!

调用方法: fun(1)
捕获异常: java.lang.ArithmeticException: / by zero
处理完毕!

调用方法: fun(2)
捕获异常: java.lang.ArrayIndexOutOfBoundsException: 5
处理完毕!

调用方法: fun(3)
捕获异常: java.lang.NumberFormatException: For input string: "56k9"
处理完毕!
```

结果除输入的“捕获异常”字符串改变外，其他异常捕捉结果均不变

该程序和上一程序的区别在于程序对异常的处理不在产生异常的fun1()方法中进行，而在哪个方法进行？
该程序中设定了哪些产生异常的方法？因此出现了3个try-catch-finally结构，哪些结构有finally语句？(3*4)
代码行：

```
package throwerror;

public class test2 {
    static void fun1(int i) throws ArithmeticException
    {
        System.out.println("调用方法: fun1("+i+")");
        if(i==0)
        {
            System.out.println("没有异常! \n");
        }
        else if(i==1)
        {
            int a=0;
            int b=10;
            b/=a;
            ArithmeticException e=new ArithmeticException();
            throw e;
        }
    }

    static void fun2() throws ArrayIndexOutOfBoundsException
    {
        System.out.println("调用方法: fun2()");
        int m[]=new int[5];
        m[5]=100;
        ArrayIndexOutOfBoundsException e=new ArrayIndexOutOfBoundsException();
        throw e;
    }

    static void fun3() throws NumberFormatException
    {
        System.out.println("调用方法: fun3()");
        String atr="56k9";
        int n=Integer.parseInt(atr);
        NumberFormatException e=new NumberFormatException();
        throw e;
    }
}
```

实验结果:

```
调用方法: fun1(0)
没有异常!

调用方法: fun1(1)
捕捉异常:/ by zero)

调用方法: fun2()
捕捉异常: java.lang.ArrayIndexOutOfBoundsException: 5

调用方法: fun3()
捕捉异常: java.lang.NumberFormatException: For input string: "56k9"
处理完毕!
```

回答:

一. 简答题

- 1
- 2
- 3

作业详情

哪个方法进行？该程序中设定了哪些产生异常的方法？因此出现了 3 个 try-catch-finally 结构，哪些结构有 finally 语句？ (3*4')
1. 异常时在 main 方法中使用多个“try-catch”块处理的；该程序对异常的处理不在产生异常的 fun（）方法中，而是将异常抛出，在主方法中进行处理。
2. fun1() 方法在（else if i=1）的情况下出现 ArithmeticException 的异常处理。
fun2（）方法在 m[5]=100 的情况下出现 ArrayIndexOutOfBoundsException 的异常处理。
fun3（）方法在将“56k9”解析为整数时可以抛出 NumberFormatException 的异常处理。
3. 所有三个 try-catch 块只有最后一个有 finally 语句。“处理完毕！”

一. 简答题

1

2

3

2. (简答题)

3. 用户自定义异常（34分）。

(1) 加法自定义异常

以下为一个自定义异常的例子，计算一定范围内的自然数之和，输入的范围值超过规定将产生异常并处理。现将计算范围修改为 (1, 10) 开区间，定义两组新范围，得到正常范围和异常范围的运行结果，代码修改2分，结果截图2分 (2*2')。

```
static int Add(int a1,int a2) throws NumRanExcep
{
    System.out.println("调用方法: Add("+a1+" , "+a2+" )");
    int sum=0;
    if(a1<1||a2>10)
    {
        NumRanExcep e=new NumRanExcep(a1,a2);
        throw e;
    }
    for(int i=a1; i<=a2; i++)
        sum+=i;
    return sum;
}
public static void main(String args[])
{
    try
    {
        int s1=Add(3,8);
        System.out.println("\t Sum="+s1+"\n");
        int s2=Add(1,20);
        System.out.println("\t Sum="+s2+"\n");
    }
    catch(NumRanExcep e)
    {
        System.out.println("\t 捕获异常: "+e.toString()+"\n");
    }
}
}
class NumRanExcep extends Exception
{
    private int i1,i2;
    NumRanExcep(int n1,int n2)
    {
        i1=n1;
        i2=n2;
    }
    public String toString()
    {
        return "NumRanExcep: "+i1+" , "+i2;
    }
}
```

程序分析

对正常范围数和异常范围数，分析代码执行异常的过程（2*3'）。

(2) 自定义异常

要求：参照上述代码，编写一个自定义异常的Application程序。功能如下：

已知两个整型数组如下：{55, 56, 24, 40, 88}和{4, 0, 8, 11}（2分），获取数组长度（2分），for循环依次读取数组元素（2分），并将两个数组打印在屏幕上（3分）。
让用户输入i和j，程序将引用这两个数组的第i个元素和第j个元素，并用第i个元素除以第j个元素（4分）。
当不能整除（2分），除数为0（2分），数组下标越界（2分）时发生异常并分别进行处理，输出自定义的异常提示字符串（1' *3分）。

我的答案：

以下为一个自定义异常的例子，计算一定范围内的自然数之和，输入的范围值超过规定将产生异常并处理。现将计算范围修改为(1,10)开区间，定义两组新范围，得到正常范围和异常范围的运行结果。

一. 简答题

1

2

3

```
public class add {
    static int Add(int a1,int a2) throws NumRanExcep
    {
        System.out.println("调用方法: Add("+a1+", "+a2+"");
        int sum=0;
        if(a1<=1||a2>=10)
        {
            NumRanExcep e=new NumRanExcep(a1,a2);
            throw e;
        }
        for(int i=a1;i<=a2;i++)
            sum+=i;
        return sum;
    }
    public static void main(String args[])
    {
        try
        {
            int s1=Add(2,3);
            System.out.println("\t Sum="+s1+"\n");
            int s2=Add(1,10);
            System.out.println("\t Sum="+s2+"\n");
        }
    }
}
```

调用方法: Add(2,3)|
Sum=5

调用方法: Add(1,10)
捕获异常: NumRanExcep:1,10

对正常范围数和异常范围数，分析代码执行异常的过程

对于正常范围数 (2, 3) :

首先在 main 方法中，Add(2, 3) 被调用，a1 被设置为 2，a2 被设置为 3。
在 Add 方法中，首先会出现结果“调用方法: Add(2,3)”。接着检查 $a1 \leq 1 \vee a2 \geq 10$ 是否成立，由于输入 (2, 3) 位于正常范围 (1, 10) 内，条件不成立，因此不会抛出异常。for 循环计算 $2 + 3$ ，结果为 5。最后结果显示“Sum=5”，没有抛出异常。

对于异常范围数 (1, 10) :

首先在 main 方法中，Add(1, 10) 被调用，a1 被设置为 1，a2 被设置为 10。
在 Add 方法中，首先会出现结果“调用方法: Add(1,10)”。然后检查条件 $a1 \leq 1 \vee a2 \geq 10$ 是否成立，由于输入 (1, 10) 超出了正常范围 (1, 10) 开区间，条件不成立，因此会抛出 `NumRanExcep` 异常。异常对象 e 被创建，包含参数 (1, 10)。异常被抛出并传递到 catch 块。在 catch 中，异常信息“NumRanExcep:1,10”显示到结果。

(2) 自定义异常

要求：参照上述代码，编写一个自定义异常的Application程序。功能如下：

已知两个整型数组如下：{55, 56, 24, 40, 88}和{4, 0, 8, 11} (2分)，获取数组长度 (2分)，for 循环依次读取数组元素 (2分)，并将两个数组打印在屏幕上 (3分)。

让用户输入 i 和 j，程序将引用这两个数组的第 i 个元素和第 j 个元素，并用第 i 个元素除以第 j 个元素 (4分)。

当不能整除 (2分)，除数为 0 (2分)，数组下标越界 (2分) 时发生异常并分别进行处理，输出自定义的异常提示字符串 (1*3分)。

代码行：

一. 简答题

1

2

3

作业详情

一. 简答题

- 1
- 2
- 3

```
public static void main(String[] args) {
    int[] array1 = {55, 56, 24, 40, 88};
    int[] array2 = {4, 0, 8, 11}; // 定义两个整型数组

    System.out.println("数组1长度: " + array1.length);
    System.out.println("数组2长度: " + array2.length); // 获取并打印数组的长度

    System.out.println("数组1内容: ");
    for (int i = 0; i < array1.length; i++) {
        System.out.print(array1[i] + " "); // for循环依次读取数组1元素
    }
    System.out.println("\n数组2内容: ");
    for (int i = 0; i < array2.length; i++) {
        System.out.print(array2[i] + " "); // for循环依次读取数组2元素
    } // 并将两个数组打印在屏幕上

    try {
        int i = 2; // 用户输入的i
        int j = 3; // 用户输入的j
        if (i >= 0 && i < array1.length && j >= 0 && j < array2.length) {
            if (array2[j] == 0) {
                throw new CustomException("除数不能为0哦"); // 当除数为零时进行处理, 输出自定义的异常提示字符串
            }
            double result = (double) array1[i] / array2[j]; // 用第i个元素除以第j个元素
            if (array1[i] % array2[j] != 0) {
                throw new CustomException("无法整除哦"); // 当不能整除时进行处理, 输出自定义的异常提示字符串
            }
            System.out.println("\n数组1[" + i + "] / 数组2[" + j + "] = " + result);
        } else {
            throw new CustomException("数组下标越界了"); // 数组下标越界时进行处理, 输出自定义的异常提示字符串
        }
    } catch (CustomException e) {
        System.out.println("\n捕获异常: " + e.getMessage());
    }
}

class CustomException extends Exception {
    public CustomException(String message) {
        super(message);
    }
}
```

实验结果:

1.除数为0时:

```
try {
    int i = 3; // 用户输入的i
    int j = 1; // 用户输入的j
    ...
} catch (CustomException e) {
    System.out.println("\n捕获异常: " + e.getMessage());
}
```

数组1长度: 5
数组2长度: 4
数组1内容:
55 56 24 40 88
数组2内容:
4 0 8 11
捕获异常: 除数不能为0哦

2.不能整除时:

```
try {
    int i = 2; // 用户输入的i
    int j = 3; // 用户输入的j
    ...
} catch (CustomException e) {
    System.out.println("\n捕获异常: " + e.getMessage());
}
```

数组1长度: 5
数组2长度: 4
数组1内容:
55 56 24 40 88
数组2内容:
4 0 8 11
捕获异常: 无法整除哦

3.数组下标越界时:

```
try {
    int i = 7; // 用户输入的i
    int j = 3; // 用户输入的j
    ...
} catch (CustomException e) {
    System.out.println("\n捕获异常: " + e.getMessage());
}
```

数组1长度: 5
数组2长度: 4
数组1内容:
55 56 24 40 88
数组2内容:
4 0 8 11
捕获异常: 数组下标越界了

实验三 异常和异常处理

一. 简答题

- 1
- 2
- 3

【目的】

- ①了解何时出现异常，常见的异常类，异常类的常用方法；
 - ②掌握异常处理的捕获异常方法和抛出异常方法；
 - ③能创建自己的自定义异常。
- 注：用户输入可用Scanner类，见第一讲《Java键盘输入》文档。

【内容】

1. 了解异常类。（34分）
- (1) 完成下列程序，类名以姓名首字母(大小不限)+学号后三位命名，分析会出现何种异常（5'）。

```
public class 自定义类名
{
    public static void main(String args[])
    {
        int i=0;

        String str[]={ "Here is an Applet. ", "Here is an Application. " };
        while(i<3)
        {
            System.out.println(str[i]);
            i++;
        }
        System.out.println("OK");
    }
}
```

程序分析

运行程序，从输出结果分析造成异常的代码为何？发生了哪种异常？由哪个异常类处理？如何修改循环结构使得程序正常运行，结果如何，附上代码和结果图。（3*4'）。

- (2) 以下是一个产生异常对象的例子。
- 以Exception_姓名首字母为类名，从程序结果分析系统如何处理异常（5分）。

```
public class _____
{
    public static void main(String args[])
    {
        Exception e;

        e=new Exception("NumberFormatException 的例子");
        System.out.println(e.toString());
        Integer i=new Integer("6a797");
        System.out.println("OK!");
    }
}
```

程序分析

运行程序，从输出结果分析产生什么异常？它由哪行代码引起？为什么会产生异常？如何修改使得程序正常运行？（3' *4）

我的答案:

- (1) 完成下列程序，类名以姓名首字母(大小不限)+学号后三位命名，分析会出现何种异常（5'）

```
package throwerror;

public class C054
{
    public static void main(String args[])
    {
        int i=0;
        String str[]={ "Here is an Applet.", "Here is an Application" };
        while(i<3)
        {
            System.out.println(str[i]);
            i++;
        }
        System.out.println("OK");
    }
}
```

作业详情

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: 2
    at throwerror.C054.main(C054.java:11)
```

出现数组越界错误，str数组只有允许两个元素，但出现str[3]时，出现了数组越界异常，为java.lang.ArrayIndexOutOfBoundsException，需要将i<3调整为i<2即可。

一. 简答题

1

2

3

运行程序，从输出结果分析造成异常的代码为何？发生了哪种异常？由哪个异常类处理？如何修改循环结构使得程序正常运行，结果如何，附上代码和结果图。（3*4）

↵

数组 str 中只有两个元素，但循环条件是 $i < 3$ ，出现了 str[3] 的越界元素，由于发生数组越界异常，会引发 ArrayIndexOutOfBoundsException 异常类。将循环条件更改为 $i < 2$ ，保证了边界不超过 2，不会引发异常。↵

```
package throwerror;

public class C054
{
    public static void main(String args[])
    {
        int i=0;
        String str[]={"Here is an Applet.", "Here is an Application"};
        while(i<2)
        {
            System.out.println(str[i]);
            i++;
        }
        System.out.println("OK");
    }
}

Here is an Applet.
Here is an Application
OK
||
```

(2) 以下是一个产生异常对象的例子。

以Exception_姓名首字母为类名，从程序结果分析系统如何处理异常（5分）。

```
package throwerror;

public class C054
{
    public static void main(String args[])
    {
        Exception e;
        e=new Exception("NumberFormatException的例子");
        System.out.println(e.toString());
        Integer i=new Integer("6a797");
        System.out.println("OK!");
    }
}

java.lang.Exception: NumberFormatException的例子
Exception in thread "main" java.lang.NumberFormatException: For input string: "6a797"
    at java.lang.NumberFormatException.forInputString(NumberFormatException.java:65)
    at java.lang.Integer.parseInt(Integer.java:580)
    at java.lang.Integer.<init>(Integer.java:867)
    at throwerror.C054.main(C054.java:10)
```

↵

产生了数据格式异常，是由于将一个字符串“6a797”转换为 Integer 对象，6a797 并非数字，将 a 去掉或改为其他数字↵

运行程序，从输出结果分析产生什么异常？它由哪行代码引起？为什么会产生异常？如何修改使得程序正常运行？（3*4）

↵

程序会引发 NumberFormatException 异常，在第四行，将一个字符串“6a797”转换为 Integer 对象。但这个字符串不是一个数字，因此会引发 NumberFormatException 异常（数据格式异常），可以使用异常处理代码或者调整字符串为数字。↵

方法一：使用异常处理代码

作业详情

```
{
    public static void main(String args[])
    {
        try {
            Exception e;
            e = new Exception("NumberFormatException的例子");
            System.out.println(e.toString());
            Integer i = new Integer("6a797");
            System.out.println("OK!");
        } catch (NumberFormatException ex) {
            System.out.println("捕获异常: " + ex.getMessage());
        }
    }
}
```

[java.lang.Exception](#): NumberFormatException的例子

捕获异常: For input string: "6a797"

方法二: 调整字符串为数字

```
package throwerror;
public class C054
{
    public static void main(String args[])
    {
        Exception e;
        e = new Exception("NumberFormatException的例子");
        System.out.println(e.toString());
        Integer i = new Integer("6797"); // 使用整数字符串
        System.out.println("OK!");
    }
}
```

[java.lang.Exception](#): NumberFormatException的例子

OK!

一. 简答题

1

2

3