

一、 关卡1

1. 训练案例1

1.1. 需求说明

1. 请说出异常的继承体系
2. 请说出你对错误(Error)的理解
3. 请说出你对编译异常(Expection的理解)
4. 请说出你对运行时异常(RuntimeException)的理解

2. 训练案例2

2.1 需求说明

请说出虚拟机处理异常的方式

3. 训练案例3

3.1. 需求说明

请说出四种运行时异常

4. 训练案例4

4.1. 需求说明

1. 请说出throw的使用位置，作用是什么？
2. 请说出 throws的使用位置，作用是什么？

5. 训练案例5

5.1. 需求说明

1. 异常处理方式有几种分别是什么
2. 详细阐述每种方式对异常是如何处理的

6. 训练案例6

6.1. 训练考核知识点

抛出异常

异常处理

6.2. 训练描述

请用代码实现如下需求

1. 写一个方法实现获取字符串某个索引上的字符的功能
2. 写一个测试方法调用上面写的方法,使用throws方式进行处理异常,并在main方法中调用这个测试方法
3. 再写一个测试方法调用上面写的方法,使用try...catch方式进行处理异常,并在main方法中调用这个测试方法

6.3. 操作步骤描述

1. 创建包com.xiaomi.level1_6
2. 定义 Test06类
3. 提供一个静态方法:public static char charAt(String str, int index) 用于获取字符串str,index索引处的字符
4. 在方法中charAt(String,int index)
 - a) 当字符串为null抛出 Exception("字符串不能为null")异常,
 - b) 当字符串为"" 的时候抛出Excetion("字符串不能为空") 异常;
 - c) 当index不在 [0,str.length()-1] 之间的时,抛出Exception("索引越界") 异常.
 - d) 否则返回str在index索引的字符
5. 提供一个静态的方法testThrows() 用于测试charAt(String,intindex)方法
 - a) 在方法中调用charAt(String,int index)方法;
 - b) 使用throws方式对异常进行处理
 - c) 在main方法中,调用本方法
6. 提供一个静态方法testTryCatch(),用于测试charAt(String,int index)方法
 - a) 在方法中调用charAt(String,int index)方法;
 - b) 使用try...catch方式对异常进行处理
 - c) 在main方法中,调用本方法.

7. 训练案例7

7.1. 训练考核知识点

自定义异常

7.2. 训练描述

请使用代码实现

1. 每一个学生(Student)都有学号,姓名和分数,分数永远不能为为负数。
2. 如果试图给学生赋值一个负数,抛出一个自定异常。

7.3. 操作步骤描述

1. 创建包com.xiaomi.level07
2. 定义异常类NoScoreException,继承 RuntimeException

a) 提供空参和有参构造方法

3. 定义学生类(Student)

a) 属性: name和score

b) 提供空参构造

c) 提供有参构造;

i. 使用setXxx方法给名称和score赋值

d) 提供setter和getter方法

i. 在setScore(intscore)方法中

1. 首先判断,如果score为负数,就抛出NoScoreException,异常信息为: 分数不能为负数:xxx.

2. 然后在给成员score赋值.

4. 定义测试类Test07

a) 提供main方法,在main方法中

i. 使用满参构造方法创建Student对象,分数传入一个负数,运行程序。

ii. 由于一旦遇到异常,后面的代码的将不在执行,所以需要注释掉上面的代码

iii. 使用空参构造创建Student对象

iv. 调用setScore(int score)方法,传入一个正数,运行程序

v. 调用setScore(int score)方法,传入一个负数,运行程序

二、 关卡2

1. 训练案例1

1.1. 训练考核知识点

自定义异常

使用代码验证知识点

继承和方法重写

使用throws的使用

1.2. 训练描述

通过代码验证

1. 方法重写, 子类只能抛比父类小的异常(父类所抛异常或父类所抛异常的子类)

a. 重写方法不能抛出与父类平级的其他异常

b. 重写方法不能抛出比父类高级的异常

c. 重写方法: 可以抛出和父类同样的异常

d. 重写方法: 可以抛出父类抛出异常的子类异常.

e. 重写方法: 可以抛出多个 "父类抛出异常的子类异常".

2. 方法重写，父类没有抛异常，子类重写父类方法也不能抛异常？

a. 父类不抛出异常,子类也不能抛出异常

b. 如果子类内部抛出了异常(或调用抛出异常的方法)只能自己处理,不允许向外抛.

注：这里所说的异常都是编译异常。

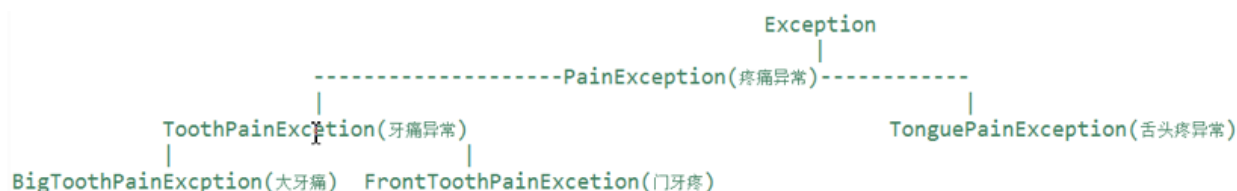
1.3. 操作步骤描述

分析:

1. 要完成上面的验证必须要一个异常体系,可以使用现有的,也可以自定义异常体系;我们这里自定义异常体系,注意异常体系中的所有异常都是编译异常;
2. 定义用于测试的父类和子类,然后逐条验证上面的说法是否正确
3. 把报编译异常的代码可以使用注释,注释起来,上面写明验证的知识点.

步骤:

1. 创建包com.xiaomi.level2_01
2. 构建异常体系



a) 定义PainException(疼异常)类继承Exception

i. 提供空参和有参构造,注意有参构造必须调用父类的有参构造

b) 定义ToothPainException(牙疼异常)类和TonguePainException(舌头疼异常)类继承PainException

i. 提供空参和有参构造,注意有参构造必须调用父类的有参构造

c) 定义BigToothPainException(大牙疼),FrontToothPainException(门牙疼)继承ToothPainException(牙痛)

i. 提供空参和有参构造,注意有参构造必须调用父类的有参构造

3. 定义用于测试的父类Father

a) 提供一个吃(eat)的成员方法，抛出牙疼异常

i. 输出吃到一个石子

ii. 抛出一个牙疼的异常

b) 提供一个喝(drink)的成员方法不抛出异常

i. 输出：喝什么都没有问题

4. 定义用于测试的子类Son,继承Father

a) 验证：重写方法不能抛出与父类平级的其他异常。

i. 重写父类的eat方法,抛出与牙疼平级舌头疼异常,编译报错,说明这句话是对的

ii. 注释掉这个重写的方法,验证下一个

b) 验证：重写方法不能抛出比父类高级的异常。

- i. 重写父类的eat方法,抛出比牙疼高级疼异常,编译报错,说明这句话是对的
- c) 验证: 重写方法可以抛出和父类同样的异常
 - i. 重写父类的eat方法,也抛出牙疼异常,编译通过,说明这句话是对的
- d) 验证: 重写方法:可以抛出父类抛出异常的子类异常.
 - i. 重写父类的eat方法,抛出牙疼异常的子类异常大牙疼异常,编译通过,说明这句话是对的
- e) 验证: 重写方法可以抛出多个 "父类抛出异常的子类异常".
 - i. 重写父类的eat方法,抛出牙疼异常的两个子类异常,大牙疼和门牙疼,编译通过,说明这句是对的
- f) 验证: 父类不抛出异常,子类也不能抛出异常(编译时异常)
 - i. 重写父类的drink方法,抛出(牙疼,舌头疼,门牙疼)任意一个编译异常,编译都报错,说明这句话是对的
- g) 验证: 如果子类内部抛出了异常(或调用抛出异常的方法)只能自己处理,不允许向外抛.
 - i. 重写父类的drink方法
 - ii. 输出喝到了100度的水
 - iii. 抛出舌头疼异常
 - iv. 这个时候编译报错
 - v. 把上面两句代码放到try代码块中
 - vi. 在catch代码块中打印异常信息
 - vii. 编译通过.

2. 训练案例2

2.1. 训练考核知识点

自定义异常类

2.2. 训练描述

请用代码描述:

在一款角色扮演游戏中,每一个人都会有名字和生命值;角色的生命值不能为负数。

要求: 当一个人物的生命值为负数的时候需要抛出自定的异常。

2.3. 操作步骤描述

1. 创建包com.xiaomi.level2_02
2. 自定义异常类NoLifeValueExption继承RuntimeException
 - a) 提供空参和有参构造
 - b) 在有参构造中, 需要调用父类的有参构造, 把异常信息传入。
3. 定义Person类
 - a) 属性: 名称(name)和生命值(lifeValue)
 - b) 提供空参构造
 - c) 提供有参构造

- i. 使用setXxx方法给name和lifeValue赋值
- d) 提供setter和getter方法
 - i. 在setLifeValue(int lifeValue)方法中
 - 1. 如果 lifeValue为负数,就抛出NoLifeValueException,异常信息为: 生命值不能为负数:xxx.
 - 2. 然后在给成员lifeValue赋值.
- 4. 定义测试类Test15
 - e) 提供main方法,在main方法中
 - i. 使用满参构造方法创建Person对象,分数传入一个负数,运行程序
 - ii. 由于一旦遇到异常,后面的代码的将不在执行,所以需要注释掉上面的代码
 - iii. 使用空参构造创建Person对象
 - iv. 调用setLifeValue(int lifeValue)方法,传入一个正数,运行程序
 - v. 调用setLifeValue(int lifeValue)方法,传入一个负数,运行程序

3. 训练案例3

3.1. 训练考核知识点

自定义异常

3.2. 训练描述

请使用代码描述

- 1. 写一个方法实现用户登陆, 传入用户名和密码
 - 2. 如果用户名错误, 就抛出自定义登陆异常(LoginException), 异常信息为用户名不存在。
 - 3. 如果密码错了就也抛出登陆异常,异常信息为密码错误
 - 4. 如果用户名和密码都对了,输出: 欢迎xxx
- 说明: 正确用户名和密码都是admin

3.3. 操作步骤描述

- 1. 创建包com.xiaomi.level2_03
- 2. 定义异常类LoginException继承Exception
 - a) 提供空参和有参构造
- 3. 创建测试类Test
 - a) 提供一个用于登陆的方法login(String name,String pwd),在放方法中
 - i. 如果用户名错误,就抛出自定义登陆异常(LoginException),异常信息为用户名不存在。
 - ii. 如果密码错了就也抛出登陆异常,异常信息为密码错误。
 - iii. 如果能来到下面,就说明用户和密码都是对的, 输出: 欢迎xxx
 - b) 提供main方法,在main方法
 - i. 使用try...catch代码块处理异常
 - ii. 调用login方法,传入错误用户名,运行程序,报运行时异常,然后注释这行代码

iii. 调用login方法,传入正确用户名,错误的命名,运行程序,报运行时异常,然后注释这行代码

iv. 调用login方法,传入正确的用户名和密码

三、关卡3

1. 训练案例1

1.1. 训练考核知识点

自定义异常

1.2. 训练描述

请使用代码描述:

每一张借记卡(DebitCard)都有账号(accountNumber)和余额(money); 余额永远不能为负数

要求: 当借记卡的余额为负数的时候需要抛出自定的异常

2. 训练案例2

2.1. 训练考核知识点

创建线程的方式

2.2. 训练描述

开启一个线程执行任务: 随机获得10个100到1000的整数并将10个整数相加求和, 将求和结果输出。