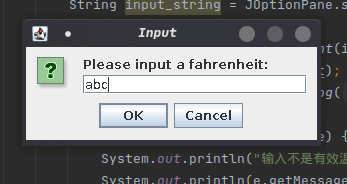
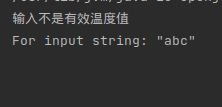
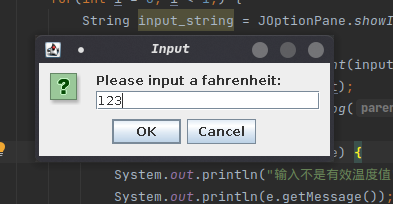
1.

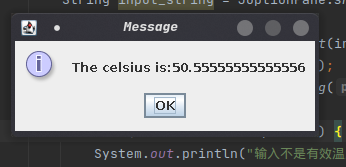
import javax.swing.\*;

public class ConvertTemperatureScale {  
 public static void main(String[] args) {  
 double fahrenheit, celsius;  
 for(int i = 0; i < 1;) {  
 String input\_string = JOptionPane.*showInputDialog*(null, "Please input a fahrenheit:");  
 try {  
 fahrenheit = Integer.*parseInt*(input\_string);  
 celsius = *convert*(fahrenheit);  
 JOptionPane.*showMessageDialog*(null, "The celsius is:" + celsius);  
 i++;  
 } catch (NumberFormatException e) {  
 System.*out*.println("输入不是有效温度值");  
 System.*out*.println(e.getMessage());  
 //e.printStackTrace();  
 } finally {  
 //System.out.println(1);  
 }  
 }  
 }  
 static double convert(double fahrenheit) {  
 return 5 \* (fahrenheit - 32) / 9;  
 }  
}



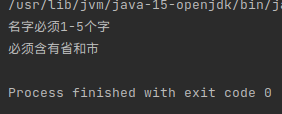






2.

public class DealError{  
 public static void main(String[] args) {  
 Student a\_student = new Student();  
 try {  
 a\_student.setName("无敌独孤不败");  
 //a\_student.setName("");  
 //a\_student.setName("无");  
 System.*out*.println(a\_student.name);  
 }  
 catch (Exception e) {  
 System.*out*.println(e.getMessage());  
 }  
 try {  
 a\_student.setAddress("北京市");  
 //a\_student.setAddress("河南省");  
 //a\_student.setAddress("河北省xx市");  
 System.*out*.println(a\_student.address);  
 }  
 catch (Exception e) {  
 System.*out*.println(e.getMessage());  
 }  
 }  
}  
  
class IllegalNameException extends RuntimeException{  
 public IllegalNameException() {  
 }  
 public IllegalNameException(String msg) {  
 super(msg);  
 }  
}  
  
class IllegalAddressException extends RuntimeException{  
 public IllegalAddressException(){  
 }  
 public IllegalAddressException(String msg) {  
 super(msg);  
 }  
}  
class Student {  
 String name, address;  
  
 void setName(String name) {  
 if(name.length() < 1 || name.length() > 5) {  
 //System.out.println(name.length());  
 throw new IllegalNameException("名字必须1-5个字");  
 }  
 else {  
 this.name = name;  
 }  
  
 }  
 void setAddress(String address) {  
 if(address.contains("省") && address.contains("市")){  
 this.address = address;  
 }  
 else {  
 throw new IllegalAddressException("必须含有省和市");  
 }  
 }  
}



3.

public class ValidTriangle {  
 double edge1, edge2, edge3;  
 double area;  
 ValidTriangle(double edge1, double edge2, double edge3) {  
 this.edge1 = edge1;  
 this.edge2 = edge2;  
 this.edge3 = edge3;  
 }  
 void getArea() {  
 if(edge1 + edge2 > edge3) {  
 double p = (edge1 + edge2 + edge3) / 2;  
 area = Math.*sqrt*(p \* (p-edge1) \* (p-edge1) \* (p-edge1));  
 System.*out*.println(area);  
 }  
 else {  
 throw new InvalidTriangle("无效三角形!");  
 }  
 }  
  
 public static void main(String[] args) {  
 ValidTriangle a\_triangle = new ValidTriangle(3, 4, 5);  
 try {  
 a\_triangle.getArea();  
 }  
 catch (Exception e) {  
 System.*out*.println(e.getMessage());  
 }  
 ValidTriangle a\_triangle1 = new ValidTriangle(2, 2, 5);  
 try {  
 a\_triangle1.getArea();  
 }  
 catch (Exception e) {  
 System.*out*.println(e.getMessage());  
 }  
 }  
}  
  
class InvalidTriangle extends RuntimeException {  
 InvalidTriangle() {  
  
 }  
 InvalidTriangle(String msg) {  
 super(msg);  
 }  
}

