1. Übung

Das folgende Programm soll die Zahlen von 1 bis 100 in eine Textdatei mit dem Namen ausgabe.txt schreiben. Leider kann das Programm in dieser Form nicht ausgeführt werden. Was ist der Grund dafür und wie können Sie es korrigieren?1 (Anmerkung: Auch wenn der Input- & Output-Stream noch nicht behandelt wurde, können Sie die Frage schon jetzt beantworten.)

import java.io.FileWriter;

public class Uebung1 {

public static void main(String[] args) {

FileWriter datei;

String text;

datei = new FileWriter("ausgabe.txt");

text = "1\n";

for(int i = 2; i <=100; i++) {

text += i;

text += "\n";

}

datei.write(text, 0, text.length());

datei.flush();

}

}

* Eventuell keine datei vorhanden
* Eventuell keine Rechte
* Eventuell Paralleler Zugriff

2. Übung In Ihrem Unternehmen beziehen Sie Autositze. Für das Beziehen stehen Ihnen die Materialien Leder und Stoff zur Verfügung. Den Stoff können Sie in jeder beliebigen Farbe liefern. Leder jedoch ist nur in den Farben Schwarz und Weiß lieferbar.

Implementieren Sie eine Klasse AutoSitze gemäß den Vorgaben des UML-Diagramms. Der Konstruktor soll eine Fehlermeldung der von Ihnen zu implementierenden Fehlerklasse FalscheParameter erzeugen, sobald er ungültige Parameterkombinationen erhält. Der Meldungstext der Fehlerklasse soll die fehlerhafte Parameterkombination ausgeben.

Testen Sie Ihre Klasse AutoSitze mit einem kleinen Testprogramm TestAutoSitzeException. In Ihrem Testprogramm soll nach erfolgreichem Durchlauf des Konstruktors eine Meldung ausgegeben werden, in welcher Farbe und in welchem Material der Sitz bezogen wurde. Sollte während dem Durchlauf des Konstruktors eine Ausnahme ausgelöst werden, geben Sie bitte in Ihrem Testprogramm den Meldungstext der Ausnahme sowie einen kleinen Hinweis, dass das Beziehen fehlgeschlagen ist, aus.

Ein Bild, das Text enthält.

Automatisch generierte Beschreibung

public class AutoSitze {  
 private boolean ledersitze;  
 private String farbe;  
  
 public AutoSitze(boolean ledersitze,String farbe) throws FalscheParameter{  
 if ((ledersitze == true) && !((farbe=="weiß") || (farbe == "schwarz"))){  
 throw new FalscheParameter(ledersitze);  
 }else {  
 this.farbe = farbe;  
 this.ledersitze = ledersitze;  
 }  
 }  
  
 public String getFarbe(){  
 return this.farbe;  
 }  
  
 public boolean istLedersitze(){  
 return this.ledersitze;  
 }  
}

public class FalscheParameter extends Exception{  
 public FalscheParameter (boolean ledersitze){  
 super("Ledersitze sind leider nur in schwarz und weiß verfügbar.");  
 }  
}

public class TestAutoSitzeException {  
 public static void main (String[] args) {  
 try {  
 AutoSitze myAutositze = new AutoSitze(false, "weiß");  
 if(myAutositze.istLedersitze()==true) {  
 System.*out*.print("Die Ledersitze sind " + myAutositze.getFarbe());  
 }else{  
 System.*out*.print("Die Stoffsitze sind " + myAutositze.getFarbe());  
 }  
 } catch (FalscheParameter e1) {  
 System.*out*.print(e1.getMessage());  
 }  
 }  
}

3. Übung Im folgenden Quellcode befindet sich ein logischer Fehler. Worin besteht er und wie kann er behoben werden?

public class TankLeerDemo {

public static void main(String[] args) {

Auto bmw = new Auto(0, 35487);

bmw.tanken();

try {

bmw.fahren();

} catch (TankLeer e) {

System.out.println(e.getMessage());

}

bmw.tanken();

try {

bmw.fahren();

} catch (Exception e) {

e.printStackTrace();

} catch (TankLeer e) {

System.out.println(e.getMessage());

System.out.println(e.toString());

e.printStackTrace();

} finally {

System.out.println("Der neue Kilometerstand: " +

bmw.getKmCount());

}

}

}

Tankt außerhalb der try blocks

Catched erst Exception und dann TankLeer