

Programmieren 1

Übungsserie Bonus

Stoff

- Bis zu Kapitel 13
- Fokus: Laufzeitfehler und die Klasse `Exception`

Allgemeine Informationen zur Abgabe

- Die Abgabe erfolgt online auf ILIAS.
- Quellcode zu den Implementationsaufgaben muss als `*.zip` Datei abgegeben werden. Exportieren Sie hierzu Ihr Projekt direkt aus Eclipse. Quellcode, den wir nicht kompilieren können, wird nicht akzeptiert.
- Arbeit in Zweiergruppen: Geben Sie jeweils nur ein Exemplar der Lösung pro Gruppe ab. Geben Sie in der Quellcode-Datei die *Namen und Matrikelnummern* beider Gruppenmitglieder in den ersten beiden Zeilen als Kommentar an.
- Vorbesprechung: 16.12.2022
- Abgabe: 23.12.2022 13:00 Uhr

Theorieaufgaben

1. Die erste der folgenden Anweisungen wird möglicherweise eine `ClassCastException`, die zweite möglicherweise eine `FileNotFoundException` werfen. Fangen Sie diese je mit einer `try-catch` Anweisung ab.

- `CDPlayer cdPlayer = (CDPlayer) mPlayer;`
- `Scanner fileScan = new Scanner(new File("/url/to/file.txt"));`

2. Die folgende Methode `increaseSemester` soll eine `NotInitializedException` werfen, falls `s` noch nicht instanziiert ist (also `null` referenziert). Ergänzen Sie die Methode so, dass die geworfene `NotInitializedException` an die aufrufende Methode zurückgegeben wird:

```
public void increaseSemester(Student s) {  
  
}
```

3. Welche Ausgabe generiert das folgende Programm:

```
class Test {  
  
    private String str = "a";  
  
    public void A() {  
        try {  
            str += "b";  
            B();  
        }  
        catch (Exception e) {  
            str += "c";  
        }  
    }  
  
    public void B() throws Exception {  
        try {  
            str += "d";  
            C();  
        }  
        catch (Exception e) {  
            throw new Exception();  
        }  
        finally {  
            str += "e";  
        }  
        str += "f";  
    }  
  
    public void C() throws Exception {  
        throw new Exception();  
    }  
  
    void display() {  
        System.out.println(str);  
    }  
  
    public static void main(String[] args) {  
        Test test = new Test();  
        test.A();  
        test.display();  
    }  
}
```

Implementationsaufgaben

1. Schreiben Sie ein Programm, mit einer eigenen `Exception` Klasse namens `StringTooLongException`, die geworfen wird, wenn eine Zeichenkette entdeckt wird, die zu viele Zeichen enthält. Lesen Sie in der `main` Methode des Programms Zeichenketten vom Benutzer ein, bis der Benutzer **"DONE"** eingibt. Wenn eine Zeichenkette eingegeben wird, die zu viele Zeichen enthält (z.B. 20), wird die Ausnahme geworfen. Die geworfene `Exception` soll das Programm beenden.
2. Ändern Sie die Lösung zu Aufgabe 1 so, dass sie die `Exception` abfängt und behandelt, wenn sie ausgelöst wird. Behandeln Sie die `Exception`,

indem Sie eine entsprechende Meldung ausgeben, und fahren Sie dann mit der Verarbeitung weiterer Zeichenketten fort.

3. Angenommen, in einem bestimmten Unternehmen erhalten alle Dokumente eine Bezeichnung, die entweder mit U, C oder P beginnt, was für "nicht klassifiziert", "vertraulich" oder "geschützt" steht. Erstellen Sie eine Ausnahmeklasse namens `InvalidDocumentCodeException`, die ausgelöst wird, wenn eine Dokumentbezeichnung eingegeben wird, die nicht zu dieser Beschreibung passt.

Erstellen Sie ein Treiberprogramm, um die Ausnahme zu testen. Behandeln Sie die Ausnahme, indem Sie eine entsprechende Meldung ausgeben, und setzen Sie dann die Verarbeitung fort.

4. Programmieren Sie eine Klasse `Dictionary`, in der man ein gegebenes deutsches Wort ins Englische übersetzen kann. Nutzen Sie hierzu die Klasse `HashMap` aus dem API.

Mit

```
HashMap<String,String> dictionary = new HashMap<String,String>();
```

können Sie eine solche Datenstruktur instanziiieren. Mit

```
dictionary.put("Luft", "Air");
```

fügen Sie Paare von Schlüssel-Werten in die Struktur ein. Weitere wichtige Methoden für die Aufgabenstellung könnten `containsKey` oder `get` sein – konsultieren Sie die Dokumentation.

Die Methode `String translate(String word)`, die Sie in der Klasse `Dictionary` programmieren, nimmt ein Wort entgegen und übersetzt dieses gemäss den gespeicherten Informationen in Ihrem `dictionary`. Falls das eingegebene Wort nicht im Wörterbuch vorhanden sein sollte, wirft Ihre Methode eine `WordNotFoundException` (diese Klasse müssen Sie selber definieren).

Schreiben Sie eine Klasse `Translator`, die eine Interaktion mit dem Benutzer nach folgendem Beispiel ermöglicht:

Geben Sie ein Wort zur Übersetzung ein: (0 um abubrechen)

Wasser

Water

Geben Sie ein Wort zur Übersetzung ein: (0 um abubrechen)

Liebe

Das Wort Liebe ist nicht im Wörterbuch vorhanden!

Geben Sie ein Wort zur Übersetzung ein: (0 um abubrechen)

Erde

Earth

Geben Sie ein Wort zur Übersetzung ein: (0 um abubrechen)

0