PR1 – Formular für Lesenotizen
WS2020/21

Nachname	Vorname	Matrikelnummer	Abgabedatum:
Abdel Kader	Schehat	1630110	19.11.2020

Dateien und Exceptions (L.6.1-L.6.3)

Lernzielfragen:

a) Worin unterscheidet sich ein Aufruf der hasNextInt-Methode eines Scanners bei Anwendung auf die Console im Vergleich zur Anwendung auf eine Datei?
- Keinen?

b) Das Statement new File("datei.txt"); führt nicht zum Anlegen einer neuen Datei. Sondern?
- Erstellt ein File Objekt, dass eine Datei repräsentiert und damit Metadaten wie Name, Länge usw. zugreifen kann, jedoch nicht auf den Inhalt. Dafür müsste man ein Scanner Objekt erstellen

c) In Ihrer main-Methode rufen Sie die folgende Methode auf:
public static void openFile throws FileNotFoundException { … }
- ➢ Was bedeutet das throws … im Methodenkopf?
  - Schlüsselwort, dass die Methode potenziell eine Ausnahme wirft, ignoriert es
  - Kann man entweder main oder eigene Methode verwenden (oder beides, wenn nötig)
- ➢ Was müssen Sie in Ihrer eigenen main-Methode programmieren, um das Programm übersetzen zu können?
  - public static void main(String[] args) throws FileNotFoundException {

d) In einer Datei namen.txt stehen Vornamen, Sie wollen wie viele mit einem x anfangen

```
import java.io.*;
import java.util.Scanner;
public class Aufd {
    public static void main(String[] args) throws FileNotFoundException {
        Scanner scanner = new Scanner(new File("names.txt"));
        int count = 0;
        String name = "";
        while (scanner.hasNext()) {
            name = scanner.next();
            if (Character.toLowerCase(name.charAt(0)) == 'x') {
                count++;
            }
        }
        System.out.println("Anzahl Namen mit X: " + count);
    }
}
```

Notizen:

File-Objekt

Methode	Beschreibung
`canRead()`	Prüft, ob Datei gelesen werden kann
`delete()`	Löscht Datei
`exists()`	Prüft, ob Datei auf dem Datenträger existiert
`getAbsolutePath()`	Gibt den Pfad im Dateisystem zurück (z. B. `"/home/stud/user/datei.txt"`)
`getName()`	Gibt den Dateinamen zurück
`isDirectory()`	Prüft, ob es sich um ein Verzeichnis handelt
`isFile()`	Prüft, ob es sich um eine Datei handelt
`length()`	Liefert die Größe der Datei in Bytes
`mkdirs()`	Erzeugt das repräsentierte Verzeichnis, falls nicht schon vorhanden.
`renameTo(`*file*`)`	Benennt die Datei um in *file*

Scanner und Dateien

Syntax: `Scanner scanner = new Scanner(new File("names.txt"));`

Ausnahmebehandlung

Exceptions/Ausnahmen ein Objekt, das einen Laufzeitfehler anzeigt

Überprüfungsbedürftige Ausnahmen (checked exceptions)

Eine Ausnahme, deren Prüfung programmiert werden muss entweder catch oder throws

Nicht überprüfungsbedürftige Ausnahmen (unchecked exceptions)

Eine Ausnahme, deren Prüfung nicht programmiert werden muss

throws

Schlüsselwort im Methodenkopf, dass aussagt, dass die Methode potenziell eine Ausnahme wirft

z.B.: `public static void main(String[] args) throws FileNotFoundException {`

Dateien schließen

Um Speicherverbrauch und den nicht Zugang zur Datei zu vermeiden mit `scanner.close();`

Token-basiertes Einlesen

Wie in Aufgabe d). Entweder Tokenanzahl bekannt => for loop, unbekannt => while loop

Berücksichtigung der Locale:

```
input.useLocale(new Locale("en", "US"));
```