

## Übung 02: Verzweigungen

Abgabetermin: xx.xx.xxxx

Name: \_\_\_\_\_

Matrikelnummer: \_\_\_\_\_

Gruppe:  G1 (Prähofer)  G2 (Wolfinger)  G3 (Wolfinger)

Aufgabe	Punkte	gelöst	abzugeben schriftlich	abzugeben elektronisch	Korr.	Pkte
Aufgabe 02.1	12	<input type="checkbox"/> <input type="checkbox"/>	Prosabeschreibung Ablaufdiagramm Java-Programm Testfälle und Ergebnisse	Java-Programm	<input type="checkbox"/>	
Aufgabe 02.2	12	<input type="checkbox"/> <input type="checkbox"/>	Prosabeschreibung Ablaufdiagramm Java-Programm Testfälle und Ergebnisse	Java-Programm	<input type="checkbox"/>	

### Aufgabe 02.1: Gleitkommazahl

Gegeben ist folgende Grammatik für die Darstellung von Gleitkommazahlen.

$$\text{Gleitkommazahl} = [ "+" | "-" ] \text{Zahl} "." \text{Zahl} [ "E" [ "+" | "-" ] \text{Zahl} ] "!"$$

Beachten Sie, dass nach der Grammatik jede Gleitkommazahl mit einem Rufzeichen abgeschlossen werden muss.

Realisieren Sie ein Programm, das Gleitkommazahlen von der Konsole einliest und überprüft, ob es sich um eine gültige Gleitkommazahl nach dieser Grammatik handelt. Verwenden Sie dazu folgende Methoden von `In`:

- `readInt()`, um eine ganze Zahl einzulesen
- `peek()`, um das nächste Zeichen anzusehen (ohne es zu lesen)
- `done()`, um zu überprüfen, ob die letzte `read`-Anweisung erfolgreich war
- `read()`, um das nächste Zeichen zu lesen.

Entwerfen Sie das Programm, indem Sie zuerst den Algorithmus in Prosa beschreiben und dann ein Ablaufdiagramm zeichnen. Realisieren Sie das Programm in Java. Achten Sie auf ansprechende Formatierung des Programms. Testen Sie das Programm mit einigen interessanten Testfällen.

Abzugeben ist:

- Die Beschreibung des Programms in Prosa
- Das Ablaufdiagramm
- Das Java-Programm
- Testfälle und die Ergebnisse.

## Lösung:

### 1) Prosabeschreibung

#### Schritt 1:

Prüfe, ob erstes Zeichen ein '+' oder '-' ist.  
Wenn ja lies dieses Zeichen ein.

#### Schritt 2:

Dann lies eine Zahl ein.  
Wenn dies in Ordnung ging dann fahre mit folgenden Schritten fort; sonst Fehler.

#### Schritt 3:

Lies das nächste Zeichen ein; dieses muss ein '.' sein; sonst Fehler.

#### Schritt 4:

Dann lies wieder eine Zahl ein.  
Wenn das in Ordnung dann fahre mit folgenden Schritten fort; sonst Fehler

#### Schritt 5:

Schau, ob nächstes Zeichen ein 'E' ist.  
Wenn ja, lies Zeichen ein, dann Schritt 6; sonst Schritt 8

#### Schritt 6:

Prüfe ob nächstes Zeichen ein '+' oder '-' ist  
wenn ja, lies dieses Zeichen ein

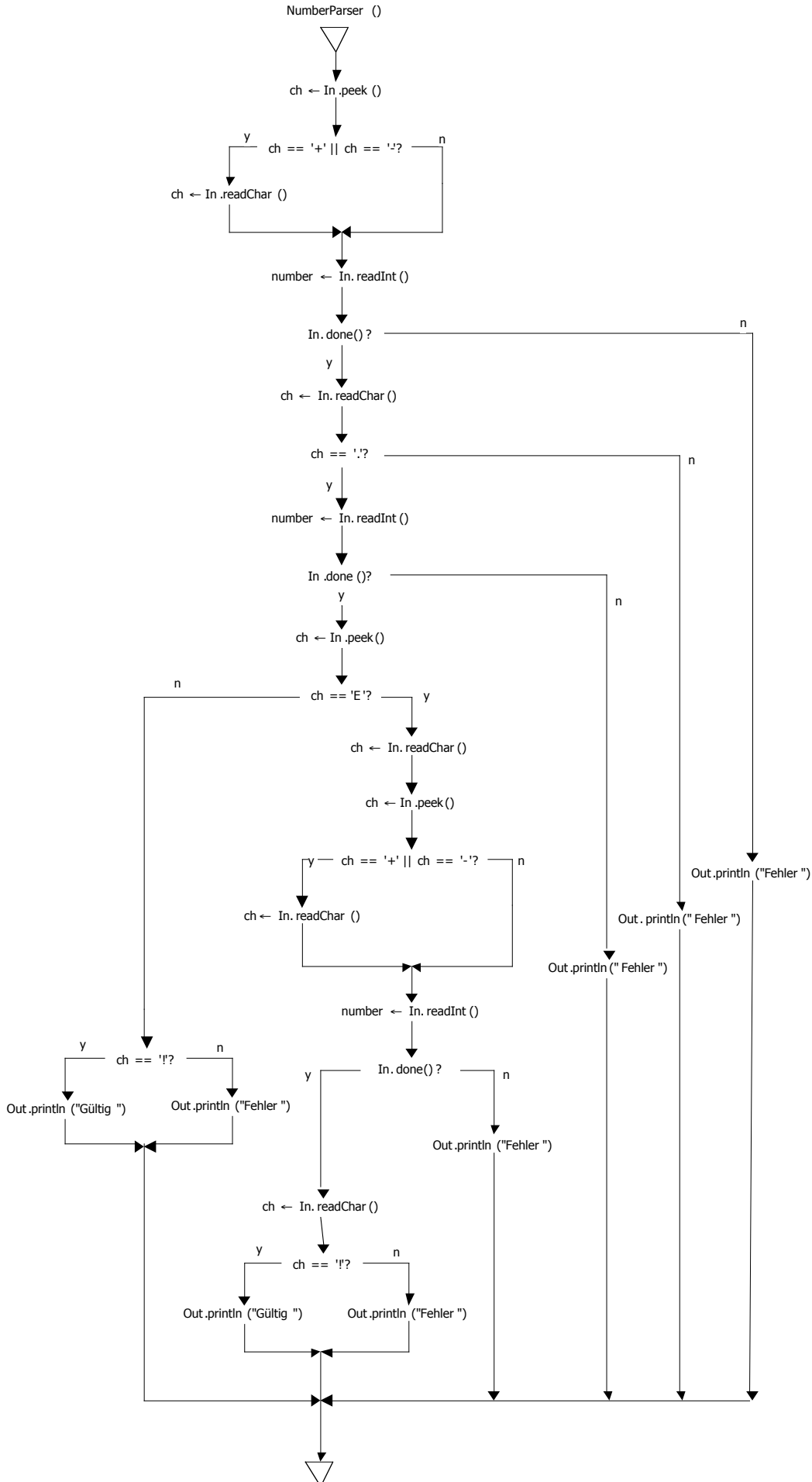
#### Schritt 7:

Dann lies wieder eine Zahl ein.  
Wenn das in Ordnung war, dann fahre mit folgenden Schritten fort; sonst Fehler

#### Schritt 8:

Lies nächstes Zeichen ein und prüfe, ob es ein '!' ist.  
Wenn ja dann Zahl richtig; sonst Fehler

2) Ablaufdiagramm



### 3) Java-Programm<sup>1</sup>

```
1 public class NumberParser {
2     public static void main(String[] args) {
3         // Schritt 1: Optionales Vorzeichen
4         char ch = In.peek();
5         if (ch == '+' || ch == '-') {
6             ch = In.readChar();
7             Out.println("-1: Vorzeichen gefunden: " + ch);
8         }
9
10        // Schritt 2: Lies Zahl ein
11        int number = In.readInt();
12        if(In.done()) {
13            Out.println("-2: Zahl gefunden: " + number);
14
15            // Schritt 3: Lies Dezimalpunkt ein
16            ch = In.readChar();
17            if(ch == '.') {
18                Out.println("-3: Dezimalpunkt gefunden: " + ch);
19
20                // Schritt 4: Lies Zahl ein
21                number = In.readInt();
22                if(In.done()) {
23                    Out.println("-4: Zahl gefunden: " + number);
24
25                    // Schritt 5: Optionaler Exponent
26                    ch = In.peek();
27                    if(ch == 'E') {
28                        ch = In.readChar();
29                        Out.println("-5: Exponent gefunden: " + ch);
30
31                        // Schritt 6: Optionales Vorzeichen
32                        ch = In.peek();
33                        if(ch == '+' || ch == '-') {
34                            ch = In.readChar();
35                            Out.println("-6: Vorzeichen gefunden: " + ch);
36                        }
37
38                        // Schritt 7: Lies Zahl ein
39                        number = In.readInt();
40                        if(In.done()) {
41                            Out.println("-7: Zahl gefunden: " + ch);
42
43                            // Schritt 8: Rufzeichen
44                            ch = In.readChar();
45                            if(ch == '!') {
46                                Out.println("-8: Rufzeichen gefunden: " + ch);
47                                Out.println("-8: Gleitkommazahl OK.");
48                            }
49                            else {
50                                Out.println("-8: Keine Gleitkommazahl. Rufzeichen nicht gefunden.");
51                            }
52                        } else {
53                            Out.println("-7: Keine Gleitkommazahl. Zahl nicht gefunden.");
54                        }
55                    } else {
56
57                        // Schritt 8: Rufzeichen
58                        if(ch == '!') {
59                            ch = In.readChar();
60                            Out.println("-8: Rufzeichen gefunden: " + ch);
61                            Out.println("-8: Gleitkommazahl OK.");
62                        }
63                        else {
64                            Out.println("-8: Keine Gleitkommazahl. Rufzeichen nicht gefunden.");
65                        }
66                    }
67                }
68            } else {
69                Out.println("-4: Keine Gleitkommazahl. Zahl nicht gefunden.");
```

```

70     }
71     }
72     else {
73         Out.println("-3: Keine Gleitkommazahl. Dezimalpunkt nicht gefunden.");
74     }
75     }
76     else {
77         Out.println("-2: Keine Gleitkommazahl. Zahl nicht gefunden.");
78     }
79     }
80 }

```

#### 4) Testfälle

#	Beschreibung	Erwartetes Ergebnis	Status
1	+10.05!	Gleitkommazahl OK	OK
2	-10.05!	Gleitkommazahl OK	OK
3	10.05!	Gleitkommazahl OK	OK
4	0.05!	Gleitkommazahl OK	OK
5	-.05!	Keine Gleitkommazahl. Zahl nicht gefunden.	OK
6	+10!	Keine Gleitkommazahl. Dezimalpunkt nicht gefunden.	OK
7	+10.!	Keine Gleitkommazahl. Zahl nicht gefunden.	OK
8	+10.05 x	Keine Gleitkommazahl. Rufzeichen nicht gefunden.	OK
9	10.05 x	Keine Gleitkommazahl. Rufzeichen nicht gefunden.	OK
10	10.05	Keine Gleitkommazahl. Rufzeichen nicht gefunden.	Fehler <sup>2</sup>
	usw.		

#### 5) Testprotokoll<sup>3</sup>

##### Testfall #1

```

+10.05!
-1: Vorzeichen gefunden: +
-2: Zahl gefunden: 10
-3: Dezimalpunkt gefunden: .
-4: Zahl gefunden: 5
-8: Gleitkommazahl OK.

```

##### Testfall #2

```

-10.05!
-1: Vorzeichen gefunden: -
-2: Zahl gefunden: 10
-3: Dezimalpunkt gefunden: .
-4: Zahl gefunden: 5
-8: Gleitkommazahl OK.

```

##### Testfall #3

```

10.05!
-2: Zahl gefunden: 10
-3: Dezimalpunkt gefunden: .
-4: Zahl gefunden: 5
-8: Gleitkommazahl OK.

```

<sup>2</sup> Programm wartet an dieser Stelle bei `In.peek()` auf weitere Eingaben. Die Eingabe kann mit der Tastenkombination `Ctrl+Z` (Windows) bzw. `Ctrl+D` (Mac OS X, Linux) abgeschlossen werden.

**Testfall #4**

```
0.05!  
-2: Zahl gefunden: 0  
-3: Dezimalpunkt gefunden: .  
-4: Zahl gefunden: 5  
-8: Gleitkommazahl OK.
```

**Testfall #5**

```
-.05!  
-1: Vorzeichen gefunden: -  
-2: Keine Gleitkommazahl. Zahl nicht gefunden.
```

**Testfall #6**

```
+10!  
-1: Vorzeichen gefunden: +  
-2: Zahl gefunden: 10  
-3: Keine Gleitkommazahl. Dezimalpunkt nicht gefunden.
```

**Testfall #7**

```
+10.!  
-1: Vorzeichen gefunden: +  
-2: Zahl gefunden: 10  
-3: Dezimalpunkt gefunden: .  
-4: Keine Gleitkommazahl. Zahl nicht gefunden.
```

**Testfall #8**

```
+10.05  
-1: Vorzeichen gefunden: +  
-2: Zahl gefunden: 10  
-3: Dezimalpunkt gefunden: .  
-4: Zahl gefunden: 5  
-8: Keine Gleitkommazahl. Rufzeichen nicht gefunden.
```

**Testfall #9**

```
10.05 x  
-2: Zahl gefunden: 10  
-3: Dezimalpunkt gefunden: .  
-4: Zahl gefunden: 5  
-8: Keine Gleitkommazahl. Rufzeichen nicht gefunden. ^
```

**Testfall #10**

```
10.05  
-2: Zahl gefunden: 10  
-3: Dezimalpunkt gefunden: .  
-4: Zahl gefunden: 5  
-8: Keine Gleitkommazahl. Rufzeichen nicht gefunden.
```