Building Java Programs

Chapter 4
Lecture 8: Scanner; if/else

reading: 3.3 – 3.4, 4.1, 4.5
A GUIDE TO
UNDERSTANDING FLOW CHARTS
PRESENTED IN FLOW CHART FORM

START

DO YOU UNDERSTAND FLOW CHARTS?

NO

OKAY. YOU SEE THE LINE LABELED "YES"?

NO

BUT YOU SEE THE ONES LABELED "NO".

NO

LISTEN.

YES

WAIT, WHAT?

NO

I HATE YOU.

GOOD

YES

AND YOU CAN SEE THE ONES LABELED "NO"?

NO

BUT YOU JUST FOLLOWED THEM TWICE!

YES

(THAT WASN'T A QUESTION.)

SCREW IT.

LET'S GO DRINK.

6 DRINKS

HEY, I SHOULD TRY INSTALLING FREEBSD!

https://xkcd.com/518/
Scanner

- **Scanner**: An object that can read input from many sources.
  - Communicates with `System.in`
  - Can also read from files (Ch. 6), web sites, databases, ...

- The **Scanner class is found in the java.util package.**

```java
import java.util.*;  // so you can use Scanner
```

- Constructing a **Scanner object to read console input**: 

  ```java
  Scanner name = new Scanner(System.in);
  ```

- **Example:**

  ```java
  Scanner console = new Scanner(System.in);  
  ```
Scanner methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>nextInt()</code></td>
<td>reads an int from the user and returns it</td>
</tr>
<tr>
<td><code>nextDouble()</code></td>
<td>reads a double from the user</td>
</tr>
<tr>
<td><code>next()</code></td>
<td>reads a one-word String from the user</td>
</tr>
<tr>
<td><code>nextLine()</code></td>
<td>reads a one-line String from the user</td>
</tr>
</tbody>
</table>

- Each method waits until the user presses Enter.
- The value typed by the user is returned.

```java
System.out.print("How old are you? "); // prompt
int age = console.nextInt();
System.out.println("You typed " + age);
```

- **prompt**: A message telling the user what input to type.
Input tokens

• **token**: A unit of user input, as read by the `Scanner`
  - Tokens are separated by *whitespace* (spaces, tabs, new lines).
  - How many tokens appear on the following line of input?
    23  John Smith  42.0  "Hello world"  $2.50  "  19"

• When a token is not the type you ask for, it crashes.

```java
System.out.print("What is your age? ");
int age = console.nextInt();
```

**Output:**

What is your age? Timmy
java.util.InputMismatchException
    at java.util.Scanner.next(Unknown Source)
    at java.util.Scanner.nextInt(Unknown Source)
...
The if statement

Executes a block of statements only if a test is true

```java
if (test) {
    statement;
    ...
    statement;
}
```

- Example:
  ```java
double gpa = console.nextDouble();
if (gpa >= 2.0) {
    System.out.println("Application accepted.");
}
```
The *if/else* statement

*Executes one block if a test is true, another if false*

```
if (test) {
    statement(s);
} else {
    statement(s);
}
```

- **Example:**
  ```java
double gpa = console.nextDouble();
if (gpa >= 2.0) {
    System.out.println("Welcome to Mars University!");
} else {
    System.out.println("Application denied.");
}
```
Relational expressions

- *if* statements and *for* loops both use logical tests.

```java
for (int i = 1; i <= 10; i++) {
    ...
}
if (i <= 10) {
    ...
}
```

- These are *boolean* expressions, seen in Ch. 5.

- Tests use *relational operators*:

<table>
<thead>
<tr>
<th>Operator</th>
<th>Meaning</th>
<th>Example</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>==</td>
<td>equals</td>
<td>1 + 1 == 2</td>
<td>true</td>
</tr>
<tr>
<td>!=</td>
<td>does not equal</td>
<td>3.2 != 2.5</td>
<td>true</td>
</tr>
<tr>
<td>&lt;</td>
<td>less than</td>
<td>10 &lt; 5</td>
<td>false</td>
</tr>
<tr>
<td>&gt;</td>
<td>greater than</td>
<td>10 &gt; 5</td>
<td>true</td>
</tr>
<tr>
<td>&lt;=</td>
<td>less than or equal to</td>
<td>126 &lt;= 100</td>
<td>false</td>
</tr>
<tr>
<td>&gt;=</td>
<td>greater than or equal to</td>
<td>5.0 &gt;= 5.0</td>
<td>true</td>
</tr>
</tbody>
</table>
Nested if/else

Chooses between outcomes using many tests

```java
if (test) {
    statement(s);
} else if (test) {
    statement(s);
} else {
    statement(s);
}
```

- Example:

```java
if (x > 0) {
    System.out.println("Positive");
} else if (x < 0) {
    System.out.println("Negative");
} else {
    System.out.println("Zero");
}
```
Nested if/else/if

- If it ends with `else`, exactly one path must be taken.
- If it ends with `if`, the code might not execute any path.

```java
if (test) {
    statement(s);
} else if (test) {
    statement(s);
} else if (test) {
    statement(s);
}
```

- Example:

```java
if (place == 1) {
    System.out.println("Gold medal!");
} else if (place == 2) {
    System.out.println("Silver medal!");
} else if (place == 3) {
    System.out.println("Bronze medal.");
}
```
Nested if structures

- exactly 1 path  \((mutually\ exclusive)\)
  ```java
  if (test) {
    statement(s);
  } else if (test) {
    statement(s);
  } else {
    statement(s);
  }
  ```

- 0 or 1 path  \((mutually\ exclusive)\)
  ```java
  if (test) {
    statement(s);
  } else if (test) {
    statement(s);
  } else if (test) {
    statement(s);
  }
  ```

- 0, 1, or many paths  \((independent\ tests;\ not\ exclusive)\)
  ```java
  if (test) {
    statement(s);
  }
  if (test) {
    statement(s);
  }
  if (test) {
    statement(s);
  }
  ```
Which nested if/else?

• (1) if/if/if  (2) nested if/else  (3) nested if/else if
  • Whether a user is lower, middle, or upper-class based on income.
    • (2)  nested if / else if / else
  • Whether you made the dean's list (GPA ≥ 3.8) or honor roll (3.5-3.8).
    • (3)  nested if / else if
  • Whether a number is divisible by 2, 3, and/or 5.
    • (1)  sequential if / if / if
  • Computing a grade of A, B, C, D, or F based on a percentage.
    • (2)  nested if / else if / else if / else if / else
Nested `if/else` question

Formula for body mass index (BMI):

\[ BMI = \frac{weight}{height^2} \times 703 \]

- Write a program that produces output like the following:

  This program reads data for two people and computes their body mass index (BMI).

  Enter next person's information:
  height (in inches)? 70.0
  weight (in pounds)? 194.25

  Enter next person's information:
  height (in inches)? 62.5
  weight (in pounds)? 130.5

  Person 1 BMI = 27.868928571428572 overweight
  Person 2 BMI = 23.485824 normal
  Difference = 4.3831045714285715

<table>
<thead>
<tr>
<th>BMI</th>
<th>Weight class</th>
</tr>
</thead>
<tbody>
<tr>
<td>below 18.5</td>
<td>underweight</td>
</tr>
<tr>
<td>18.5 - 24.9</td>
<td>normal</td>
</tr>
<tr>
<td>25.0 - 29.9</td>
<td>overweight</td>
</tr>
<tr>
<td>30.0 and up</td>
<td>obese</td>
</tr>
</tbody>
</table>