

Übung Java switch Expression

Ausgangslage

Bei der Übung ifelse haben Sie einen Code dem Http Code Range zugewiesen. In dieser Übung soll nun der exakte Text zu einem HTTP-Code ausgegeben werden. Hierzu ist der HTTP-Code als Kommandozeilenparameter zu übergeben. Das HTTP-Protokoll definiert die folgenden Successful-Codes mit Text: 100 Continue 200 OK 201 Created 202 Accepted 203 Non-Authoritative Information 204 No Content 205 Reset Content 206 Partial Content. Das HTTP-Protokoll definiert die folgenden Redirection-Codes mit Text: 300 Multiple Choices 301 Moved Permanently 302 Moved Temporarily 303 See Other 304 Not Modified 305 Use Proxy. Das HTTP-Protokoll definiert die folgenden Client-Error-Codes mit Text: 400 Bad Request 401 Unauthorized 402 Payment Required 403 Forbidden 404 Not Found 405 Method Not Allowed 406 Not Acceptable 407 Proxy Authentication Required 408 Request Timeout 409 Conflict 410 Gone 411 Length Required 412 Precondition Failed 413 Request Entity Too Large 414 Request-URI Too Long 415 Unsupported Media Type. Das HTTP-Protokoll definiert die folgenden Server-Error-Codes mit Text: 500 Internal Server Error 501 Not Implemented 502 Bad Gateway 503 Service Unavailable 504 Gateway Timeout 505 HTTP Version Not Supported. Kommandozeilenparameter werden beim Aufruf dem Programm übergeben. Die Parameter werden mit einem Space (Blank) separiert und im Array `args[]` der Methode `main(...)` abgelegt. Das folgende Codefragment zeigt die Auswertung und Anzeige von Kommandozeilenparametern:

```
public static void main (String[] args) {
    for (int i=0; i < args.length; i++) {
        System.out.println("#34;param[#34; + i + #34;]: #34; + args[i]);
    }
}
```

Beim Ausdruck `args[i]` handelt es sich um einen String. Dieser String ist nun in einen primitiven `int`-Datentyp zu konvertieren. Das folgende Codefragment zeigt diese Konversion über die Wrapper-Klasse `Integer` auf:

```
int number = Integer.parseInt (args[i]);
```

Vorgehen

Lösen Sie bitte die Aufgabe wie folgt: Erzeugen Sie die Datei `HttpCodeText.java` mit einem geeigneten Editor. In Eclipse eröffnen Sie eine neue Klasse am einfachsten über das Menu `New->Class`: Geben Sie jeder Klasse neben dem Namen ein Package mit. Programmieren Sie die Klasse `HttpCodeText` mit dem Gerüst der Methode `main(String []args)` aus und programmieren Sie die Klasse weiter gemäss dem folgenden Listing:

```
public class HttpCodeText {
    public static void main(String[] args) {
        if (args.length < 1) {
            System.err.println("#34;no argument specified#34;);
            return;
        }
        for (int i = 0; i < args.length; i++) {
            System.out.println("#34;param[#34; + i + #34;]: #34; + args[i]);
        }
    }
}
```

Speichern Sie das Programm ab und führen Sie solches aus via Popup Menu `Run As->Java Application`: Leider ergibt das Programm keine Ausgabe an die Konsole, weil keine Kommandozeilen Parameter definiert sind. Öffnen Sie nun über das Menu `Run->Run Configurations` den folgenden Dialog: Definieren Sie wie im Dialog die Werte `#34;200 302 404 500 -1 600#34;` und lassen Sie das Programm erneut ausführen: Die Konsole sollte die Werte nun korrekt anzeigen: Nun geht es darum, innerhalb der Schleife die Argumente in eine `int`-Variable umzuwandeln und dem Http Code den korrekten Text zuzuordnen via `switch` Statement. Das folgende Listing zeigt einen möglichen Ansatz:

```
for (int i = 0; i < args.length; i++) {
    System.out.println("#34;param[#34; + i + #34;]: #34; + args[i]);
    int httpCode = Integer.parseInt(args[i]);
    String text = "#34;#34;";
    switch(httpCode) {
        case 200: text = "#34;OK#34;"; break;
        ... TODO ...
        default: text = "#34;Unknown#34;"; break;
    }
    System.out.println(httpCode + #34;: #34; + text);
}
```

Programmieren Sie einige der möglichen Varianten aus. Zusatzaufgabe: Was passiert bei der Eingabe eines ungültigen Http Codes wie z.B. `#34;eins#34;` ausgeschrieben?

Lösung

Eine mögliche Lösung finden Sie hier

Kontakt

Simtech AG
Finkenweg 23
3110 Münsingen

Schweiz

Impressum

Das Copyright für sämtliche Inhalte dieser Website liegt bei Simtech AG, Schweiz.
Beachten Sie auch unsere Hinweise zum Urheberrecht, Datenschutz und Haftungsausschluss.
Jeder Hinweis auf Fehler nehmen wir gerne entgegen.

Copyright

2024 Simtech AG, All rights reserved, Powered by stack.ch written in Golang by Daniel Schmutz

<https://www.simtech-ag.ch/java-11-schulung-kurs-8-einführung-ressourcen-switch>