Übung HTTP Code Map

Ausgangslage

Bei der Übung Switch wurde jedem HTTP Code die richtige Fehlermeldung zugewiesen. Diese Zuordnung wurde mit einem switch-Statement gelöst. Mit dieser Übung soll die gleiche Anwendung nun über eine HashMap in Form der Klasse java.util.HashMap gelöst werden. Hierzu ist als Key-Objekt der HTTP-Code zu verwenden. Das Value-Objekt soll dem zugeordneten Text entsprechen.Das HTTP-Protokoll definiert die folgenden Successful-Codes mit Text:100 Continue200 OK201 Created202 Accepted203 Non-Authoritative Information204 No Content205 Reset Content206 Partial ContentDas HTTP-Protokoll definiert die folgenden Redirection-Codes mit Text:300 Multiple Choices301 Moved Permanently302 Moved Temporarily303 See Other304 Not Modified305 Use ProxyDas HTTP-Protokoll definiert die folgenden Client-Error-Codes mit Text:400 Bad Request401 Unauthorized402 Payment Required403 Forbidden404 Not Found405 Method Not Allowed406 Not Acceptable407 Proxy Authentication Required408 Request Timeout409 Conflict410 Gone411 Length Required412 Precondition Failed413 Request Entity Too Large414 Request-URI Too Long415 Unsupported Media TypeDas HTTP-Protokoll definiert die folgenden Server-Error-Codes mit Text:500 Internal Server Error501 Not Implemented502 Bad Gateway503 Service Unavailable504 Gateway Timeout505 HTTP Version Not SupportedKommandozeilenparameter 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 ("param[" + i + "]: " + 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

- Erzeugen Sie ein neues Java Projekt oder ein Package für diese Übung.

- Erzeugen Sie die Klasse HttpCodeMap
- .

- Programmieren Sie die Klasse HttpCodeMap mit dem Gerüst der Methode main(String []args) aus.

- Werten Sie den Kommandozeilen-Parameter aus, der den Fehlercode beinhaltet.
Hinweis:Überprüfen Sie auch den Fall, dass kein Parameter vorhanden ist.

- Instanziieren Sie nun ein Objekt vom Type java.util.HashMap und fügen Sie alle Http Codes in die Table ein. Das Key Objekt soll dem Http Code (Wrapper Klasse java.lang.Integer) und das Value-Objekt dem Text ensprechen.

- Holen Sie nun zum gegebenen Http Code den entsprechenden Text aus der HashMap und zeigen Sie diesen an.

- Falls der Http Code nicht existiert, so soll dies entsprechend mitgeteilt werden.

Zusatzaufgabe Print Entries

- Geben Sie alle Map Entries mit der for-each Schleife aus.

- Geben Sie alle Map Entries mit der forEach Lambda Funktion aus.

Zusatzaufgabe Tree Map

- Kopieren Sie die Lösung um in die Klasse HttpCodeTreeMap und ersetzen Sie die HashMap mit der TreeMap.

- Was stellen Sie fest.

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 Haftungsauschluss.
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/kurse-jpf2-http-code-map