

情報科学プロジェクト実験レポート課題

S142063 佐藤涼亮

平成 28 年 12 月 6 日

CGI(Common Gateway Interface)

1 課題の内容

CGI プログラムによる WEB インターフェースでのデータベース操作

1.1 要点

- C++ プログラムを CGI プログラムに修正
- 第 9 回目のレポート課題の内容を WEB インターフェースで使えるように変更

2 プログラムの説明

- ・郵便番号から住所を検索するプログラム
- ・住所から郵便番号を検索するプログラム

前回に引き続き以上の 2 つのプログラムを Web のインターフェースで使えるように修正・変更した。CGIinput.hpp は研究室独自のクラスであり、メソッドが GET と POST の両方に対応しており、同一のプログラムとして実装できる。また、フォームデータを解析して容易に使用できるように実装されている。

2.1 目的

Web のインターフェースにおけるデータベースの操作。

2.2 方法

CGI プログラムによるデータベースを操作するプログラムの実装。

2.3 結果

Web インターフェースで実行した結果をまとめる。

初期状態

検索システム

住所検索

検索したい住所、または郵便番号を入力してください。

郵便番号: ex.123-4567 → 1234567

住所 : ex.東京都〇〇区△△△

検索結果

郵便番号と住所を入力

検索システム

住所検索

検索したい住所、または郵便番号を入力してください。

郵便番号: ex.123-4567 → 1234567

住所 : ex.東京都〇〇区△△△

検索結果

search をクリックすると検索結果が表示される

検索システム

住所検索

検索したい住所、または郵便番号を入力してください。

郵便番号: ex.123-4567 → 1234567

住所 : ex.東京都〇〇区△△△

検索結果

郵便番号: 1790084 の住所は 東京都練馬区氷川台 です。

住所 : 東京都小笠原村母島 の郵便番号は 1002211 です。

郵便番号のみを存在しない値で入力

検索システム

住所検索

検索したい住所、または郵便番号を入力してください。

郵便番号: ex.123-4567 → 1234567

住所 : ex.東京都〇〇区△△△

検索結果

郵便番号: 1790084 の住所は 東京都練馬区氷川台 です。

住所 : 東京都小笠原村母島 の郵便番号は 1002211 です。

入力されなかった住所は何も表示せず
入力された郵便番号は一致するものがないので

検索システム

住所検索

検索したい住所、または郵便番号を入力してください。

郵便番号: ex.123-4567 → 1234567

住所 : ex.東京都〇〇区△△△

検索結果

郵便番号: 一致するものはありません

2.4 考察

郵便番号や住所の検索を行うことができ、期待通りの結果が得られた。

3 感想

CGI プログラムについて学習した。今回は C++ の言語から作成したが、CGI プログラムは、いろんなプログラミング言語で作成することができるので、少し知識のある perl でも今後扱ってみたい。そして、情報検索の授業で学習したと組み合わせて、Web 検索 CGI プログラムを作成したい。また、以前の研究室の課題でも HTML/CSS の学習をしたが、最近、自主的に HTML/CSS の学習を行っているため、HTML と今回の課題は関連性も高く、深く理解することができた。HTML/CSS の知識を活かし、もっと使用者のことを考え、見やすく扱いやすいように修正してみたい。

4 プログラム

ソースコード 1: report.cpp

```
1 #include <iostream>
2 #include <MyDBS3.hpp>
3 #include <CGIinput.hpp>
4 using namespace std;
5
6 // エラー処理
7 int error(string e,string str){
8     e += str;
9     cout << e + "</body></html>";
10    return 1;
11 }
12
13 int main()
14 {
15     CGIinput fmd; // 宣言とともに入力を終える
16
17     // 結果を出力
18     string s = "Content-type: text/html\r\n\r\n";
19 }
```

```

20 // HTML を返す時のタイプ
21 s += R"(
22 <!doctype_html><html>
23 <head><meta_charset="utf-8"><title>report</title></head>
24 <body><h1_style="background-color: #F2F2E5;">検索システム</h1>
25 <h2>住所検索</h2>
26 <p>検索したい住所、または郵便番号を入力してください。</p>
27 <form_action="db.cgi"_method="post">
28 <p>郵便番号:<input_name="post"_size="20"/>ex.123-4567<input_type="text" value="1234567"></p>
29 <p>住所:<input_name="addr"_size="20"/>ex.東京都 区<input_type="text" value=""></p>
30 <input_type="submit" name="OK" value="search">
31 </form>
32 <h3_style="border-bottom: solid 1px #000000;">検索結果</h3>
33 )";
34
35 // データベースファイルを開く
36 MyDBS d("db/tokyo.db");
37 if (!d) return error(s, "db_error");
38
39 string sql;
40
41 // 郵便番号での検索
42 if (!fmd["post"].empty()){
43     sql = "select_kanji1,kanji2,kanji3_from_post_where_num=?";
44     if (d.prepare(sql,fmd["post"]) != SQLITE_OK){
45         return error(s,d.error());
46     }
47     string kanji1,kanji2,kanji3;
48     if (d.step(&kanji1,&kanji2,&kanji3) != SQLITE_ROW){
49         s += "<p>郵便番号:<input_type='text' value='<input_name='post' type='text' value='<input_name='post' type='text' value='";
50     }
51     else{
52         s += "<p>郵便番号:<input_type='text' value='";
53         + kanji1 + kanji2 + kanji3 + "<input_type='text' value='";
54         while (d.step(&kanji1,&kanji2,&kanji3) == SQLITE_ROW){
55             s += "<p>郵便番号:<input_type='text' value='";
56             + kanji1 + kanji2 + kanji3 + "<input_type='text' value='";
57         }
58     }
59 }
60
61 // 住所での検索
62 if (!fmd["addr"].empty()){
63     sql = "select_num_from_post_where_kanji1||kanji2||kanji3=?";
64     if (d.prepare(sql,fmd["addr"]) != SQLITE_OK){
65         return error(s,d.error());
66     }
67     string num;
68     if (d.step(&num) != SQLITE_ROW){
69         s += "<p>住所:<input_type='text' value='";
70     }
71     else{
72         s += "<p>住所:<input_type='text' value='";
73         + fmd["addr"] + "<input_type='text' value='";
74         while (d.step(&num) == SQLITE_ROW){
75             s += "<p>住所:<input_type='text' value='";
76             + fmd["addr"] + "<input_type='text' value='";
77         }
78     }
79     s += "</body></html>";
80     cout << s;
81
82     return 0;
83 }

```