

Hotelli C++

1. Kirjastot

```
4  #include <iostream>
5  #include <vector>
6  #include <string>
7  #include <cstdlib>
8  #include <ctime>
9  #include <iomanip>
10 #include <algorithm>
11 #include <cctype>
12
13 using namespace std;
```

Mahdollistavat tehtävien suorittamiseen vaaditut toiminnot

2. Globaalit muuttujat

```
15 const int MIN_HUONEITA = 40;
16 const int MAX_HUONEITA = 300;
17 const int MIN_HINTA = 80;
18 const int MAX_HINTA = 100;
19 const int VARAUSNUMERO_MIN = 10000;
20 const int VARAUSNUMERO_MAX = 99999;
```

Eivät muutu koodin suorittamisen aikana, määritellään huoneiden lukumäärä, hinta ja mahdollinen varausnumeron lukuväli.

3. Apufunktiot

```
22 string trim(const string& str) {
23     size_t start = str.find_first_not_of(" \t");
24     size_t end = str.find_last_not_of(" \t");
25
26     return (start == string::npos || end == string::npos) ? "" : str.substr(start, end - start + 1);
27 }
28
29
30 string toLower(const string& str) {
31     string lowerStr = str;
32     transform(lowerStr.begin(), lowerStr.end(), lowerStr.begin(),
33         [](unsigned char c) { return tolower(c); });
34
35     return lowerStr;
36 }
37
```

Poistaa merkkijonoista ylimääräiset välilyönnit sekä sarkaimet, muuttaa merkkijonot pieniksi kirjaimiksi, pienentää vikaherkkyyttä koodissa ja yksinkertaistaa varauksen hakuprosessia.

4. Structit

```
39 struct Huone {
40     int numero;
41     int koko;
42     double hinta;
43     bool varattu;
44 };
45
46
47
48 struct Varaus {
49     int varausNumero{};
50     string varaajanNimi;
51     int huoneNumero{};
52     int oidenMaara{};
53     double loppuSumma{};
54 };
55
56
57
```

Määritellään muuttujat varauksille, joihin tallennetaan varauksen tiedot.

5. Pääohjelma (Main)

```
69  int main() {
70
71      srand(static_cast<unsigned>(time(0)));
72
73      vector<Huone> huoneet = alustaHuoneet();
74      vector<Varaus> varaukset;
75
76      int valinta;
77
78      do {
79
80          naytaValikko();
81          cin >> valinta;
82          cin.ignore();
83
84          switch (valinta) {
85
86              case 1:
87                  teeVaraus(huoneet, varaukset);
88                  break;
89
90              case 2:
91                  haeVaraus(varaukset);
92                  break;
93
94              case 3:
95                  cout << "Kiitos, että käytit varausjärjestelmää. Näkemiin!" << endl;
96                  break;
97
98              default:
99
100                 cout << "Virheellinen valinta! Yritä uudelleen." << endl;
101
102             }
103
104         } while (valinta != 3);
105
106         return 0;
107     }
```

Alustetaan satunnaislukugeneraattori ja alustetaan tietorakenteet, alustetaan vektorit tyhjäksi. Näytetään käyttäjälle käyttöliittymä, joka käsitellään switch-rakenteella.

6. Huoneet

```

110 vector<Huone> alustaHuoneet() {
111     int huoneidenMaara = MIN_HUONEITA + rand() % ((MAX_HUONEITA - MIN_HUONEITA) / 2) * 2;
112     vector<Huone> huoneet;
113     for (int i = 1; i <= huoneidenMaara; ++i) {
114         Huone huone;
115         huone.numero = i;
116         huone.koko = (i <= huoneidenMaara / 2) ? 1 : 2;
117         huone.hinta = MIN_HINTA + rand() % (MAX_HINTA - MIN_HINTA + 1);
118         huone.varattu = false;
119         huoneet.push_back(huone);
120     }
121     return huoneet;
122 }
123
124
125
126
127
128

```

Luodaan hotellihuoneet ja määritellään ominaisuudet, eli numero, koko, hinta ja varaus.

7. Varaus ja alennus

```

130 int arvoVarausNumero() {
131     return VARAUSNUMERO_MIN + rand() % (VARAUSNUMERO_MAX - VARAUSNUMERO_MIN + 1);
132 }
133
134
135
136
137 double laskeAlennus(double loppuSumma) {
138     int alennusProsentti = (rand() % 3) * 10;
139     return loppuSumma * (1 - alennusProsentti / 100.0);
140 }
141
142

```

Arvotaan käyttäjälle varausnumero sekä alennus satunnaislukugeneraattorilla.

8. Varausten käsittely

```

143
144 void teeVaraus(vector<Huone>& huoneet, vector<Varaus>& varaukset) {
145
146     cout << "Anna varaajan nimi: ";
147     cin.ignore();
148     string nimi;
149     getline(cin, nimi);
150
151     string siistiNimi = trim(nimi);
152
153     cout << "Valitse huoneen koko (1 = yhden hengen, 2 = kahden hengen): ";
154
155     int koko;
156     cin >> koko;
157
158     cout << "Anna oiden maara: ";
159
160     int yot;
161     cin >> yot;
162
163     int huoneNumero = -1;
164
165     for (auto& huone : huoneet) {
166
167         if (!huone.varattu && huone.koko == koko) {
168             huoneNumero = huone.numero;
169             huone.varattu = true;
170
171             break;
172         }
173     }
174
175     if (huoneNumero == -1) {
176
177         cout << "Valitettavasti ei löytynyt vapaita huoneita kyseiselle huonetyypille." << endl;
178
179         return;
180     }
181
182     double loppuSumma = huoneet[huoneNumero - 1].hinta * yot;
183     double alennettuSumma = laskeAlennus(loppuSumma);
184
185     Varaus varaus;
186     varaus.varausNumero = arvoVarausNumero();
187     varaus.varaajanNimi = siistiNimi;
188     varaus.huoneNumero = huoneNumero;
189     varaus.oidenMaara = yot;
190     varaus.loppuSumma = alennettuSumma;
191
192     varaukset.push_back(varaus);
193
194     cout << "Huone varattu onnistuneesti!" << endl;
195     cout << "Varausnumero: " << varaus.varausNumero << endl;
196     cout << "Huone: " << huoneNumero << ", Oiden maara: " << yot << endl;
197     cout << "Loppusumma: " << fixed << setprecision(2) << alennettuSumma << " euroa" << endl;
198 }
199

```

```

200 bool onkoHuoneVapaa(vector<Huone>& huoneet, int huoneNumero) {
201     if (huoneNumero < 1 || huoneNumero > static_cast<int>(huoneet.size()))
202         return false;
203     return !huoneet[huoneNumero - 1].varattu;
204 }
205
206
207
208
209 void haeVaraus(const vector<Varaus>& varaukset) {
210     cout << "Haluatko hakea (1) varausnumerolla vai (2) nimellä? ";
211
212     int valinta;
213     cin >> valinta;
214     cin.ignore();
215
216     if (valinta == 1) {
217         cout << "Anna varausnumero: ";
218         int numero;
219         cin >> numero;
220
221         for (const auto& varaus : varaukset) {
222             if (varaus.varausNumero == numero) {
223                 cout << "Varaus löytyi!" << endl;
224                 cout << "Nimi: " << varaus.varaajanNimi << ", Huone: " << varaus.huoneNumero
225                     << ", Oita: " << varaus.oidenMaara << ", Hintaa: " << fixed << setprecision(2)
226                     << varaus.loppuSumma << " euroa" << endl;
227                 return;
228             }
229         }
230         cout << "Varausta ei löytynyt varausnumerolla." << endl;
231     }
232     else if (valinta == 2) {
233         cout << "Anna varaajan nimi: ";
234         cin.ignore();
235         string nimi;
236         getline(cin, nimi);
237
238         string hakunimi = toLower(trim(nimi));
239
240         for (const auto& varaus : varaukset) {
241             string tallennettuNimi = toLower(trim(varaus.varaajanNimi));
242             if (tallennettuNimi == hakunimi) {
243                 cout << "Varaus löytyi!" << endl;
244                 cout << "Nimi: " << varaus.varaajanNimi << ", Huone: " << varaus.huoneNumero
245                     << ", Oita: " << varaus.oidenMaara << ", Hintaa: " << fixed << setprecision(2)
246                     << varaus.loppuSumma << " euroa" << endl;
247                 return;
248             }
249         }
250         cout << "Varausta ei löytynyt nimellä." << endl;
251     }
252     else {
253         cout << "Virheellinen valinta." << endl;
254     }
255 }
256
257
258
259
260
261
262
263
264
265
266
267
268
269

```

Kysytään käyttäjältä varaajan nimi, siistitään tämä nimi trim-funktiolla kuten aikaisemmin mainittu, tallennetaan huoneen koko sekä öiden määrä. Etsitään vapaa huone ”huoneet”-vektorista tai ilmoitetaan, ettei sellaista ole. Lasketaan kokonaiskustannukset ja tallennetaan varaus ”Varaus”-rakenteeseen, joka lisätään ”varaukset”-vektoriin. Tulostetaan nämä tiedot käyttäjälle varauksen yhteydessä sekä erikseen kysyttäessä. Varauksen tietoja kysyttäessä käytetään getline()-komentoa.

9. Valikko

```

270
271 void naytaValikko() {
272
273     cout << "\n*****" << endl;
274     cout << "      HOTELLIN VARAUSJARJESTELMA      " << endl;
275     cout << "*****" << endl;
276     cout << "1. Tee uusi varaus" << endl;
277     cout << "2. Hae varaus" << endl;
278     cout << "3. Poistu ohjelmasta" << endl;
279     cout << "*****" << endl;
280     cout << "Valitse toiminto (1-3): ";
281 }
282

```

Näytetään graafinen valikko käyttäjälle ohjelman käynnistyessä.

(Tässä kohtaa loppui aika Microsoft Forms valikon tekemiseen, eikä dokumentointi ole vahvin osaaminen itsellä.)