



C++ Ohjelmoinnin perusteet

Nuutti Ruuskanen

RAPORTTI
Joulukuu 2024

Ajoneuvotekniikka

23AutoB

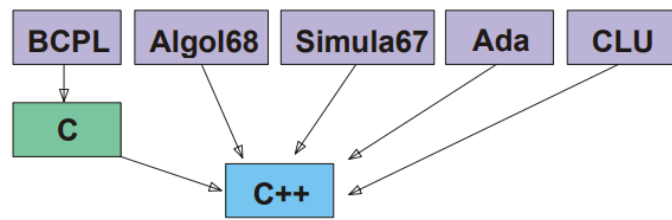
2307704

SISÄLLYS

1	Johdanto	3
2	Ajatusympyrä	4
3	Funktiot/avainsanat.....	5
3.1	Main	5
3.2	Void.....	5
3.3	Struct.....	6
3.4	Caset.....	7
3.5	Int.....	8
4	Kirjastot.....	9
4.1	<iostream>	9
4.2	<string>	9
4.3	<vector>	9
4.4	<iomanip>	9
4.5	<limits>.....	9
4.6	<algorithm>	9
5	POHDINTA	10
	LÄHTEET.....	11

1 Johdanto

Kurssin tavoitteena on oppia c ohjelmointikielen C++ varianttia. Ohjelmointikieli on saanut alkunsa 1970 luvulla, kehittäjänä on Dennis Ritchie. Kielen kehittämiseen kului melkein vuosikymmeniä, ja kieleen saatiin ANSI standardi kesällä 1983. Kieli on yhdistelmä useampaa ohjelmointikieltä, jota tuohon aikaan käytettiin. Suurimpina vaikuttajina kielessä on C-kieli ja bcpl (Basic Combined Programming language). (teku.fi/C_Kieli)



Kuva 1-1. C- ja C++ kielten perustana olevat kielet.

Kuva 1. C++ kielen rakenne

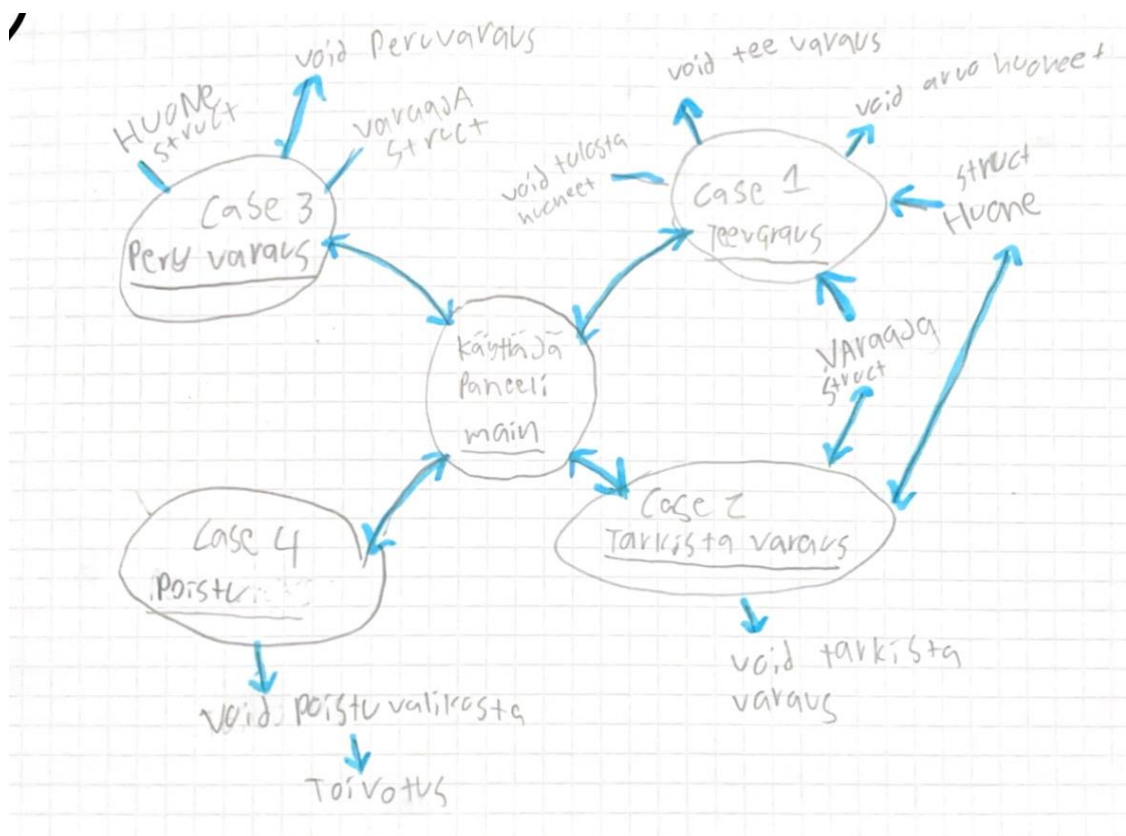
Ohjelmointi oli minulle henkilökohtaisesti täysin uutta, ja en ole itse ikinä harrastanut mitään, joka vaatisi tietokoneen käytön osaamista. Käsitykseni ohjelmoinnista oli myös hyvin erilainen kurssin ohjelmointi tehtäviin verrattuna, sillä ajattelin että ohjelmointi tapahtuu pelkillä numeroilla. Kuten mainittiin kurssin tavoitteena, on opetella käyttämään ohjelmakieltä erilaisten yksinkertaisten toimintojen tekemiseen, esimerkiksi numeroiden arpomiseen. Kurssilla tehdään kotitehtäviä, sekä isompi lopputyö hotellin huoneenvaraus ohjelma, jota tämä raportti käsittelee.

Lähdin kurssille opettelemaan yleisimmän ohjelmointikielen perusteita, koska omalla alallani älykkäät koneet puolella tulen tarvitsemaan myöhemmin ohjelmointia, joten tämä on hyvä "valmisteleva" kurssi tulevia ohjelmointi projekteja varten.

2 Ajatusympyrä

Ohjelman tarkoitus tiedetään, mutta kurssilla opetettiin, että kun ohjelma koostuu useammasta eri ohjelmasta (pää ja ali), niin kannattaa hahmotella ohjelmaa paperille, esimerkiksi tässä työssä käyttäjäpaneeli, josta käyttäjä valitsee esimerkiksi huoneen varauksen, on keskeisessä osassa ohjelmaa, joten piirsin käyttäjäpaneelin keskeiseen ympyrään. Tämä helpottaa miettimään mitä aliohjelmiin halutaan lisätä, ja mitä halutaan laittaa muistiin.

Ohjelma on hotellin varauskalenteri, joten mietin myös, mitä hotellissa asioidessa tapahtuu, kun varataan huonetta. Esimerkiksi kysytään millaisen huoneen varaisi, montako majoittujaa. Asiakkaalta myös yleensä kysytään henkilötietoja, kuten puhelinnumeroa ja koti osoitetta.



Kuva 2. Ohjelmani ajatusympyrä.

Kuvasta poiketen structien nuolet pitäisi vetää void aliohjelmiin.

3 Funktiot/avainsanat

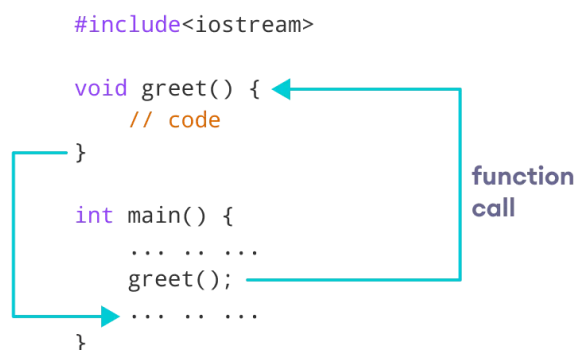
3.1 Main

Main funktio on tarpeellinen jokaisessa C++ kielen ohjelmassa. Mainia voidaan käyttää sellaisenaan, tai sitä voidaan käyttää pääohjelmana, kuten olen ohjelmassani tehnyt. Main funktion tarvitsee aina palauttaa arvonsa, joka ei onnistu ilman int funktiota. Main funktiosta voidaan poistua Retun komennolla.

Pääohjelmaan rakensin "failsafen", eli jos käyttäjä syöttää jonkun kuin numeron 1-4 ohjelma alkaa virheilmoituksen, ja pyytää käyttäjää syöttämään luvun uudelleen. Tämä on toteutettu if komennolla, hauskan kommentin kanssa. Lohkoon oli pakko liittää vielä `cin.clear` ja `cin.ignore(numeric_limits<streamsize>::max())`. Näillä komennoilla tyhjennetään väärästä merkistä johtuva virhetila ja ignore tyhjentää koko syötteen, jotta ohjelma ei kaadu. Ilman näitä komentoja ohjelma kaa-
tuu niin, että ohjelma pyytää loputtomalla loopilla numeroa uudelleen.

3.2 Void

Void funktiota käytettiin kurssilla yleensä aliohjelmana. Voidilla voidaan suorittaa erilaisia tehtäviä mainin ulkopuolella. Funktio tarvitsee pyynnön, esimerkki main funktiosta, jotta ohjelma osaa siirtyä aliohjelmaan. Void funktiot ovat sijoiteltuna aina main funktion yläpuolelle. Void funktoilla ei palauteta arvoja mainiin, vaan niihin siirrytään. Voidista poistutaan komennolla exit.

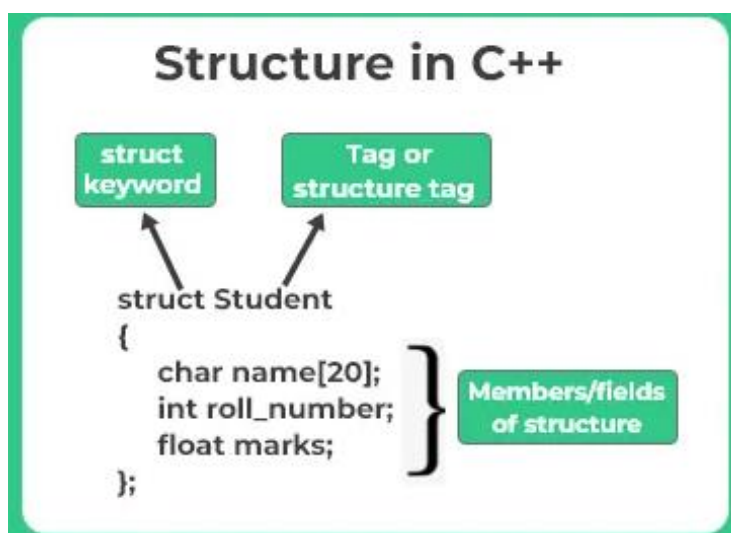


Kuva 3. Voidin toiminta

Käytin Void funktiota ohjelmassani, koska funktion käyttö tekee ohjelmasta mielestäni selkeämmän ja helpommin kirjoitettavan. Ohjelmani toiminnot, kuten *tulostaHuoneet()* tai *peruVaraus()*, eivät tarvitse laskettua arvoa käytettäväksi muualla, joten paluuarvoa ei tarvittaisi. Void funktiot voivat käsitellä tietorakenteita, kuten globaalia *hotellinHuoneet*-vektoria, suoraan ilman ylimääräistä arvojen siirtelyä. Lisäksi ne helpottavat ohjelman käytettävyyttä tarjoamalla tulokset, kuten varaustiedot suoraan näytölle.

3.3 Struct

Struct on lyhenne sanasta structure. Rakennetta käytin ohjelmassani tietopankkina. *Varaaja*-struct tallentaa varaajan tiedot, kuten nimen, puhelinnumeron ja kotiosoitteen. *Huone*-struct sisältää huoneen tiedot, kuten numeron, varaustilan ja alennuksen. Structien käyttö mahdollistaa tapauksessani kaikkien hotellin huoneiden hallinnan yhdestä vector säilöstä. Structeista saan tiedot esimerkiksi henkilötiedot varauksen tarkistukseen.



Kuva 4. Struct rakenne.

3.4 Caset

Main funktioon sijoitin case rakenteen. Mielestäni tämä on ohjelmani paras ominaisuus, sillä se helpotti ainakin itseäni paljon ohjelman hahmottelussa, koska sijoitin caseihin ali ohjelmat, jotka oli nimetty hotellivirkailijan kanssakäytävän asian mukaan. Rakenne on nimeltään switch case, jossa caset eli tilanteet on numeroitu käyttäjäpaneelin alkutekstin mukaisesti, eli kun käyttäjä valitsee numeron 1, niin case 1 valitaan ja *varaaHuone* aliohjelma aukeaa. Tämä selkeyttää mielestäni paljon.

```
switch (valinta) {  
  case 1:  
    varaaHuone();  
    break;  
  case 2:  
    tarkistaVaraus();  
    break;  
  case 3:  
    peruVaraus();  
    break;  
  case 4:  
    poistuValikosta();  
    break;  
}
```

Kuva 5. Case esimerkki ohjelmastani.

3.5 Int

Ohjelmointikielessä int on lyhenne sanasta integer tämä on datatyyppi C++ ohjelmointi kielessä. Tällä int funktiolla voidaan tallentaa numeroita, esimerkiksi main funktion sisälle. Integer siis tallentaa numeron, joka palautetaan mainiin.

Funktio soveltuu tähän käyttötarkoitukseen, koska käyttäjän ei tarvitse valita desimaali lukuja, vaan tasalukuja, jotka voivat olla positiivisia tai negatiivisia. Funktiolla saadaan tallennettua numerot väliltä (-2 147 483 648) – (2 147 483 647). Main funktioni aloitetaan myös int funktiolla.

Integerejä käytin ohjelmassani esimerkiksi opiskelija alennuksen ja huoneen määrän tallentamiseksi. Käytin sitä vielä muun muassa main funktiossa palvelun valinnassa, joka ohjaa ohjelman toimintaa,

```
void arvoHuonenumerot() { // Tämä aliohjelma arpoo huonenumeron ja alennuksen.  
  
    for (int i = 1; i <= 17; ++i) { // Määritetään hotelliin 17 huonetta.  
        Huone uusiHuone;  
        uusiHuone.numero = rand() % 150 + 1; // Arvotaan huonenumerot välille 1-150.  
        uusiHuone.onYhdenhuone = (i <= 7); // Asetataan 7 ensimmäistä huonetta yhden hengen huoneiksi.  
        uusiHuone.opiskelijaAle = 7 + rand() % 11; // Arvotaan alennus välille 7-17%.
```

Kuva 5. Esimerkki int funktion käytöstä ohjelmassani.

4 Kirjastot

Tehtävässä käytin useita eri kirjastoja, eli `#include <>` komentoja. Kirjastoilla mahdollistetaan useiden eri toimintojen tekeminen ohjelmassa, kuten funktoiden toiminta. Kaikki funktiot eivät toimi samalla kirjastolla, vaan jokaiselle on oma käyttökohteensa.

4.1 `<iostream>`

Tämä kirjasto on välttämätön lähes jokaisen ohjelman toiminnan kannalta, sillä tällä kirjastolla mahdollistetaan lukeminen ja kirjoittaminen, eli `cin` ja `cout`. Lukemista ja kirjoittaminen on tärkeää tässä ohjelmassa.

4.2 `<string>`

String kirjastolla saadaan tallennettua kirjaimia tai numeroita. Tarvitsemme tämän kirjaston, koska ohjelmassa on käytössä aliohjelmia, jotka sisältävät string komentoja.

4.3 `<vector>`

Kirjastolla saadaan rakennettua taulukoita, esimerkiksi ohjelmassani on `vector<string>` kanssamatkustajat. Kirjasto mahdollistaa kanssamatkustajan lisäämisen. Ja string tallentaa sen. Näitä kutsutaan globaaleiksi muuttujiksi.

4.4 `<iomanip>`

Kun halutaan käyttää desimaaleja tarvitaan `iomanip` kirjasto. Tämä kirjasto mahdollistaa `setprecision` funktion käyttämisen, esimerkiksi huoneen lopullisen hinnan ilmoittamiseen.

4.5 `<limits>`

Ohjelmassa on käytössä komento `double` kohdassa hinnan lasku. Limits kirjasto tarvitaan tämän tiedoston käsittelyyn.

4.6 `<algorithm>`

Käytössä on komento `all_off`, jolla tarkastellaan puhelin numeron sopivuutta (tarvitsee olla numereellinen). Tämä komento tarvitsee `algorithm` kirjaston.

5 POHDINTA

Ohjelmaani jäi reilusti parannettavaa. Päällimmäisenä parannuksena mieleeni se, että jos ohjelma arpoo samat huoneen numerot yhden ja kahden hengen huoneelle, niin ohjelma valitsee aina yhden hengen huoneen. Yritin korjata ohjelman if/else komennolla, eli if huoneissa on samat numerot, niin else antaa valita kumman huonetyypin haluaa, mutta en saanut tätä jostain syystä toimimaan.

Mielestäni ohjelmani olisi arvosanan 3 arvoinen. Ohjelmassa on elementtejä 4-5 arvoisesta työstä, mutta mielestäni ohjelmani toimivuuden selittämisessä on aukkoja. Mielestäni sisäistin paremmin ohjelman kirjoituksen, kuin sen, miten ohjelman eri komponentit syvällisemmin selitettynä oikein toimivat. Siksi annoin itselleni arvosanaksi 3.

LÄHTEET

C++ Kurssin perusteet kurssi materiaali. Viitattu 1.12.2024

https://web.archive.org/web/20150217001408/http://teku.fi/eBooks/eBook_01_CKieli/C_Kieli_01.pdf Yleistietoa C ohjelmointi kielestä. Viitattu 8.12.2024

https://www.w3schools.com/cpp/cpp_variables.asp C++ variables selitettynä. Viitattu 9.12.2024

<https://docs.mql4.com/basis/types/integer/integertypes> Int funktion numeraalinen rajoite. Viitattu 9.12.2024

<https://prepinsta.com/c-plus-plus/structures/> Kuva 4. Struct rakenne. Viitattu 9.12.2024