

Hotellihuoneen varausjärjestelmä – harkkatyö

Mikko Rajamäki

Harjoitustyössäni toteutin hotellihuoneen varausjärjestelmän, joka käyttää kurssilla opittuja ohjelmoinnin peruskäsitteitä ja rakenteita. Ohjelma on suunniteltu vastaamaan arvosanan 3 vaatimuksia. Tässä on tiivistetty kuvaus työn toiminnallisuudesta ja käytetyistä ohjelmointimenetelmistä.

1. Ohjelman toiminta

Ohjelma arpoo hotellin huonemäärän (30–70) ja hinnan per yö (80–100 euroa) käyttäen satunnaislukuja. Käyttäjä voi varata huoneita syöttämällä huonenumero ja öiden määrän. Ohjelma tarkistaa, onko valittu huone jo varattu, ja laskee varausten loppusumman. Varaaminen jatkuu, kunnes käyttäjä päättää lopettaa tai kaikki huoneet on varattu.

2. Kurssilla opittujen asioiden soveltaminen

Ohjelmassa hyödynnettiin seuraavia kurssilla opittuja käsitteitä ja rakenteita:

1. Muuttujat ja vakioarvot:

- Käytin selkeitä ja kuvaavia muuttujanimiä, kuten `huoneidenMaara` ja `hintaPerYo`, sekä vakioita (esim. `MIN_HINTA` ja `MAX_HINTA`) ohjelman perusparametrien määrittelyyn.

2. Satunnaislukujen generointi:

- `std::srand` ja `std::rand` -funktiolla ohjelma arpoo huonemäärän ja huonehinnan.

3. Taulukot ja vektorit:

- Käytin C++-standardeja `std::vector`-vektoreita tallentamaan huoneiden varausstatukset.

4. Ohjelman jakaminen aliohjelmiin:

- Toteutin kaksi aliohjelmaa:
 - `tarkistaHuoneVarattu`: Tarkistaa, onko huone varattu.
 - `laskeHinta`: Laskee loppusumman käyttäjän syöttämän yömääriin perustuen.

- Näin ohjelmakoodi pysyy selkeänä ja modulaarisena.

5. Valinta- ja toistorakenteet:

- if- ja while-rakenteita käytettiin käyttäjän syötteen tarkistamiseen ja ohjelman toistuvaan toimintaan varausten aikana.

6. Virheenkäsittely:

- Käytin `std::cin.clear` ja `std::cin.ignore` varmistaakseni, että käyttäjän syötteet käsitellään oikein, ja estin ohjelmaa kaatumasta virheellisiin syötteisiin.

7. Käyttäjäystävällisyys ja vuorovaikutus:

- Ohjelmassa on selkeät ohjeet ja ilmoitukset, jotka kertovat käyttäjälle, mitä tehdä seuraavaksi.