

# Programmeerimine keeles C++

## Praktikum 6: C++ lisateegi kasutamine: võrguteek RakNet.

### Üldised nõuded

Selle praktikumi koduse tööna peate edasi arendama aine veebilehel antud alusprogrammi. Töötav `Makefile` on olemas, teie ülesanne on lahendada ülesanne ja dokumenteerida selle klassid. Ülesande lahendamiseks vajalikele materjalidele on viited praktikumi veebilehel.

RakNet teegi leidmiseks vajalikud viited leiate aine veebilehelt. Alusprogrammi kompileerimiseks RakNet teegiga peate `Makefile` alguses määrama muutujas `RAKNETDIR` teegi juurkausta. Lisaks, peate vastavalt oma operatsioonisüsteemile eemaldama kommentaari ühe järgneva rea eest (kehtib MinGW ja Linuxi kasutajatele).

### Ülesanne 1 – Klientprogramm arvu äraarvamise mängu jaoks

Alusprogrammis on antud valmis server, mis suudab teenindada arvu ära arvavat klienti. Ära tuleb arvata arv vahemikus 0-99 (mõlemad kaasa arvatud). Lisaks on antud poolik klientprogramm. Teie ülesandeks on klientprogramm valmis kirjutada. Selleks peate tegema vähemalt järgmised sammud:

- 1) Lisage `GuessingClient` klassile meetod `void SendGuessMessage (unsigned short guess)`, mis saadab serverile sõnumi kliendi pakkumisega. Paketi identifikaatoriks pange päises defineeritud `PACKET_GUESS`.
- 2) Lisage `GuessingClient` klassi meetodis `NextFrame()` `PACKET_GUESS_REPLY` paketi töötlemine.
- 3) Lisage meetod `void ProcessGuessReplyMessage (Packet* packet)`, mis eelmises punktis nimetatud paketti töötleb, lugedes sealt välja vastuse kliendi pakkumisele.
- 4) Täiendage peaprogrammi (failis `ClientApp.cpp`) nii, et kasutajalt küsitaks käsureal pakkumisi, kuni mõni neist osutub õigeks. Kõik pakkumised saadetakse serverile, kutsudes peaprogrammist välja meetodit `SendGuessMessage`. Serveri vastust töötletakse meetodis `ProcessGuessReplyMessage`. Lisage `GuessingClient` klassile muutujaid ja meetodid, mille abil peaprogramm saab teada, et serverilt on vastus saabunud ning samuti vastuse.
- 5) Kui pakkumine on õige, peaks klient väljastama pakkumiste arvu, mis õige arvu äraarvamiseks kulus.

Tulemusena peab valmima programm, mille käivitamisel klient ühendub serveriga ning seejärel alustab kasutajalt arvude küsimist. Õige arvu äraarvamisel lõpetab programm töö. Pange tähele, et

te ei tohiks serveri funktsionaalsust muuta, sest testimisel käivitan ma originaalse koodiga serveri ning eeldan, et teie klient oskab selle külge ühenduda ja sellega edukalt suhelda. Leidub lahendus, mis kasutab päises `GuessingGame.h` kirjeldatud konstante ning töötab `GuessingServer.cpp` koodiga töötava serveriga. Lahenduse leidmine nõuab mõlema programmi ning teegi tundmaõppimist ning vajalike täienduste tegemist.

Lahenduse eest saab kuni 9 punkti. Kõige olulisem on see, et programm töötaks ning sellega saaks kirjeldatud mängu mängida. Mittetoimiva koodi puhul võin paluda selle täiendamist nii kaua kuni programm edukalt tööle läheb.

### **Lisaülesanne 1 – Iseseisvalt arvu ära arvav klientprogramm**

Kirjutage klientprogramm, mis suudab ilma kasutaja abita serveriga suheldes arvu ära arvata. Lahenduse eest saab 1 lisapunkti, kui ühegi arvu äraarvamine ei võta üle kaheksa korra (ja see number on isegi varuga, teoorias saab ka seitsmega).

### **Lisaülesanne 2 – Mängija nimi kliendis ja serveris**

Lisage `PACKET_INTRODUCTION` sõnumisse mängija nimi ning laske serveril seda kasutada väljundis tema IP-aadressi asemel (loomulikult pärast seda, kui nimi on serverini jõudnud). Lahenduse eest saab 1 lisapunkti. Ühe täiendava lisapunkti saab lahendus, kus on serveris kiiremini äraarvanute TOP 10 edetabel. Pange tähele, et selle ülesande lahendamiseks peate serverit muutma.

### **Tähelepanu – küsimused ülesande kohta ja muud tingimused**

Võttes arvesse, et ülesanne nõuab tugevalt iseseisvat tööd, siis julgustan teid muredega puhul listi kirjutama.

Materjalid ja lahenduste üleslaadimise süsteem on aine veebilehel. Tähtaeg on praktikumi toimumisnädala pühapäeva õhtu kell 23:59 (aega on 7 päeva).