

IAS0410 Object-orienteeritud programmeerimine

Kursuse eesmärgid

Antud kursuse eesmärgiks on anda kuulajatele sellises mahus teadmisi ja praktilise oskusi, et nad oleksid pärast kursuse lõpetamist võimelised alustama tööd algaja tarkvaraarendajana.

Eeldused

Kohustuslikke eeldusaineid sellel kursusel ei ole. Üldiselt siiski eeldatakse, et üliõpilased on õppinud programmeerimist C keele baasil. Neil, kelle programmeerimise oskused baseeruvad keeltele C# või Java, tuleb kõvasti pingutada, kuid nad peaksid hakkma saama. Pythoni tundmisest on paraku väga vähe abi. Ja loomulikult pole see kursus neile, kes üldse ei ole kunagi programmeerimist õppinud.

Varustus

Kursusel kasutatav operatsioonisüsteem on Windows ning programmeeriskeel C++. Kuna osa tööd tuleb nagunii teha väljaspool auditooriumi, peab igal üliõpilasel olema arvuti, millel jookseb:

- Windows (versioonid 10 või 11). Kursusetöös kasutatakse tarkvara, mis Linux ja Mac arvutitel ei tööta.
- Microsoft Visual Studio arenduskeskkond C/C++ toega (2019 või 2022 aasta versioon). Vabavarast *Visual Studio Community Edition* piisab täielikult (<https://visualstudio.microsoft.com/downloads/>).
- Kes soovivad lõpetada kursuse hindega “suurepärase” (5), peavad alla laadima ka Qt arenduskeskkonna *Qt non-commercial open-source version*. See on samuti vabavara (<https://www.qt.io/download-open-source>).

Aeg ja koht

Kursust peetakse ainult sügissemestril. Tunnid (kord nädalas, 4 * 45 minutit) hõlmavad nii loenguid kui ka harjutusi..

Klassiruumiks on ICT-121 / 122. Tunnid toimuvad igal kolmapäeval 16:00 – 19:15.

Keel

Õppetöö toimub eesti keeles. Õppematerjalid on inglise keeles.

Teemad

Kuivõrd eestikeelsed terminid tihtipeale puuduvad või ei ole üldtunnustatud, siis arusaamatuste vältimiseks on kursuses käsitletavate teemade loetelu kirjutatud inglise keeles.

1. Introduction to C++:
 - a. Memory allocation operators.
 - b. Unicode and type *wchar_t*. Operations with Unicode strings.
 - c. Scope resolution operator.
 - d. Default values for function arguments.
 - e. Overloading.
 - f. Inline functions.

- g. References. *lvalue* and *rvalue*.
 - h. Exceptions
2. Object-oriented programming fundamental paradigms in C++:
- a. Objects and classes.
 - b. Declaration of classes and definition of member functions.
 - c. Creating of objects and working with them.
 - d. Access modifiers and accessor functions.
 - e. Constructors.
 - f. Destructors.
 - g. Inline member functions.
 - h. Aggregation
 - i. Inheritance
 - j. Protected members and deriving modes.
 - k. Polymorphism, virtual functions and late binding.
3. Deeper into C++:
- a. Structs in C++.
 - b. Copy constructor.
 - c. Pointer *this*.
 - d. Friends.
 - e. Operator overloading.
 - f. Assignment overloading.
 - g. Static members.
 - h. Constant members.
 - i. Constant objects
 - j. *Static*, *reinterpret*, *const* and *dynamic* casts.
 - k. New keywords: *bool* and *nullptr*.
 - l. Namespaces and the *using* directive.
 - m. C++ standard library.
 - n. Input / output streams. Global objects *cin* and *cout*.
 - o. File streams. Operations with text files and binary files.
 - p. Standard exceptions.
 - q. Strings and operations with them.
 - r. Stringstreams.
4. Advanced C++:
- a. Class templates and function templates.
 - b. New keyword *auto*.
 - c. Operator *typeid*.
 - d. Complex numbers.
 - e. *Variant*, *any* and *optional* templates
 - f. Initializing and constructor initializer.
 - g. Pointers to functions and to class member functions.
 - h. Lambda expressions.
 - i. Function wrappers.
 - j. Functors.
 - k. Smart pointers
 - l. Rational numbers
 - m. Standard classes for time handling.
 - n. Standard classes for random number generation.

5. Containers and algorithms:
 - a. Vectors and iterating.
 - b. Lists.
 - c. Initializer lists.
 - d. Range-based *for*-loop.
 - e. Variants.
 - f. Pairs.
 - g. Maps.
 - h. Hashing and unordered maps
 - i. Bitsets
 - j. Searching (*find*, *find_if*).
 - k. Algorithm *for_each*.
 - l. Filling (*generate*, *fill*)
 - m. Copying (*copy*, *copy_n*, *copy_if*).
 - n. Replacing (*replace*, *replace_if*).
 - o. Removing (*remove*, *remove_if*).
 - p. Special (*unique*, *sort*)
6. Concurrency:
 - a. Multithreading: organization and problems. Race conditions.
 - b. Launching, temporary stopping and ending of threads.
 - c. Mutexes and locks.
 - d. Atomic variables.
 - e. Polling.
 - f. Conditional variables.
 - g. Producer-consumer problem.
 - h. Futures.
 - i. Promises.
 - j. Asynchronous input / output in Windows.
 - k. DLLs, explicit and implicit linking.
7. Qt framework:
 - a. Installation
 - b. QtCreator and selecting of kits.
 - c. Creating of simple Qt widget applications with Qt designer.
 - d. Signals and slots.
 - e. Events.
 - f. Qt threads.
 - g. Signals through thread boundaries.
 - h. Qt thread synchronization.
 - i. Qt strings and containers.
 - j. Qt input / output devices. File handling and remote device operations.
 - k. Setting parameters storing and retrieving.
 - l. Layouts.
8. C++ in Microsoft .NET environment
 - a. The main concepts of .NET: assembly, managed code, native code, JIT compiler.
 - b. .NET language family.
 - c. Data types in .NET.
 - d. Properties, classes, structures and delegates.
 - e. C++/CLI.

Kursusetöö

Kursusetööst täieliku ülevaate saamiseks tuleb alla laadida [spetsifikatsioon](#). Allpool esitatakse lühike kokkuvõte.

Õppejõu poolt on ette valmistatud [DLL](#), mis emuleerib mingit kujuteldavat keemiatehast. Tehas on varustatud kujuteldavate anduritega, mis mõõdavad temperatuuri, rõhku reaktoris jne. DLL väljastab mõõtmistulemusi (tegelikult juhuslikke suursi) juhuslikel ajamomentidel.

Kursusetöö ülesandeks on kirjutada mitmelõimeline (multithreaded) C++ programm, mis lingib enda külge ülalnimetatud DLLi, loeb saadetavaid mõõtmistulemusi, näitab neid ekraanil ja salvestab etteantud andmestruktuuri ning sealt kettafai.

Üliõpilane võib valida kahe variandi vahel:

1. Lihtsama variandi puhul kujutab väljatöötatav programm endast konsooli (command prompt window) kaudu kommuniqueeruvat aplikatsiooni. Käsud programmi juhtimiseks sisestatakse klaviatuurilt ja tulemusi näidatakse konsooli aknas. Selle variandi puhul on maksimaalseks võimalikuks hindeks “väga hea” (4).
2. Keerulisema variandi puhul on programm varustatud Qt arenduskeskkonnas väljatöötatud graafilise kasutajaliidesega. Sellise programmi eest on võimaloik saada hinnet “suurepärase” (5).

DLL võib emuleerida kümnet erinevat tehast. Programmi sisemise andmestruktuuri jaoks on 4 erinevat deklaratsiooni. Üliõpilane leiab oma variandi vastavast [tabelist](#).

Kirjandusest

C++ keeles programmeerimise kohta on kirjandust väga palju ning põhimõtteliselt võib igaüks leida endale sobiva raamatu. Probleem on selles, et raamatukogudes leidub neid parimal juhul vast üks või kaks eksemplari, e-poodides (näiteks Amazon.de või Amazon.se) on hinnad aga kõrged.

Kõige põhjalikumalt käsitlevad C++ programmeerimist 2 autorit: Marc Gregoire ja Nicolai M. Josuttis. E-poodidest võib leida nende raamatute loetelu.

Interneti teatmikud:

- <http://msdn.microsoft.com/en-US/>
- <http://www.cplusplus.com/reference/>
- <https://doc.qt.io/>

Head nõu saab tihtipeale lehelt <https://www.geeksforgeeks.org/c-plus-plus/>.

[Powerpointi slaidid](#) ja [näidisprojektid](#) peaksid sisaldama kogu kursuses käsitletavat materjali.

Hindamine

Eksamit kui sellist ei ole. Hinne antakse kursusetöö põhjal. Hindamise reeglid on [kursusetöö spetsifikatsiooni](#) lõpus.

Konsultatsioonid

Probleemide puhul kirjutage viktor.leppikson@liewenthal.ee