C++

Robotex 2014

TTÜ Robotiklubi

Tallinn University of Technology Robotics Club

Arrayd, massiivid

Array - eesti keeles massiiv

Indekseeritakse 0-st.

Kindla suurusega massiivi tekitamine

```
int viis_numbrit[5];
viis_numbrit[0] = 1;
```

Kui vaja minev mälu hulk pole teada koodi kirjutamise hetkel, siis saab kasutada

```
int palju_numbreid[] = new int[kui_palju];
```

Mälu vabastamine

Kasutatud mälu peab ka vabastama, kui seda ei tee siis lekib programm mälu.

Kellele sellised programmid meeldiks?

Mälu vabastatakse delete[] operaatoriga.

```
delete[] palju_numbreid;
```

Ei ole vaja ette anda, palju mälu vabastada tuleb.

Mäluga käsitsi tegelemine on ohtlik, võib tekkida vigu.

Parem on kasutada olemasolevaid (teiste poolt tehtud) vahendeid.

vector

vector<int>, vector<float> jne

Tekst

C keeles on teksti hoidmiseks char massiivid. Suurem osa vigu tekivad nende kasutamisega, seepärast on C++ keeles string tüüp.

```
string nimi;
cout << "Mis su nimi on?" << endl;
cin >> nimi;
cout << "Tere, " << nimi << endl;
```

Ohutum kasutada, ei teki probleeme lõpust üle minemisega, ei pea kontrollima, kas lõpus on 0 või mitte. Kui vaja teha stringist tavaline C stiilis char massiiv siis on olemas meetod

c_str();

map

Võimaldab hoida andmeid paaridena võti ja väärtus

Nagu vector, võimaldab hoida erinevat liiki andmeid.

Võrreldav teiste keelte hash tüübiga (aga siin saab võtmeks olla suvaline tüüp)

```
map<int, string> bussid;
bussid[18] = "Laagri";
bussid[5] = "Männiku-Kose";
```

Map kasutamine

map

```
for (map<int, string>::iterator it = bussid.begin();
     it != bussid.end(); it++) {
    cout << "buss nr " << (*it).first << " sõidab" <<
            (*it).second << endl;
}
```

it on siin std::pair tüüpi objekt, kus mapi võti on "first" ja andmed "second" muutujas.

Failidest lugemise ja kirjutamisega tegelevad järgmised failid

päisefailid

#include <ifstream> - lugemine (i - input)

#include <ofstream> - kirjutamine (o - output)

#include <fstream> mõlemad

lugemise

koodinäide

```
#include <iostream>
#include <string>

using namespace std;
int main(int argc, char *argv[])
{
    ifstream fail;
    fail.open("nimed.txt");
    string tekst;
    if (!fail) {
        cout << "Faili ei avatud" << endl;
        return -1;
    }

    while(!fail.eof()) {
        fail >> tekst;
        cout << tekst << endl;
    }
    return 0;
}
```

```
ifstream fail;
fail.open("fail.txt")
char rida[1000];
fail.getline(rida, 1000);
```

Luuakse 1000 chari jaoks massiiv, antakse ifstream.getline() meetodile koos massiivi pikkusega. getline loeb failist kuni rea lõpuni või kuni 1000 tähemärki loetud on ja tulemuse paneb muutujasse 'rida'.

Debugimine, silumine

Breakpoint - koht kus debugger programmi seisma paneb ja laseb uurida programmi seisu, muutujate väärtusi jne.

F10 - step into - astub aktiivsel real oleva funktsiooni väljakutse sisse.

F11 - step over - astub üle aktiivsel real oleva funktsiooni väljakutse.

Tõnis Märtmaa

e-mail: robotiklubi@robotiklubi.ee

tel. +372 53408660