TTÜ Robotiklubi

Tallinn University of Technology Robotics Club

TTÜ Robotiklubi

Tallinn University of Technology Robotics Club

Kursuse eesmärgid

- C++ keele alused
- Objektorienteeritud programmeerimine
- Koodi organiseerimine ja head kombed
- Qt

Loodi 1979. aastal. C keele edasiarendus. Lisatud OOP võimalusi.

C on raualähedane. C++ võimaldab kirjutada koodi kõrgemal tasemel.

Keerulisemate süsteemide programeerimine, kus on jõudlus tähtis - operatsioonisüsteemid, arvutimängud, Eestis e-valimised.

Uusim versioon C++11

Lihtne endale jalga lasta :)

C++ keel, ajalugu, kus kasutatakse?

24.03.14

```
#include <iostream> // see siin on kommentaar
/*see on
ka
kommentaar
*/

int main (int argc, char *argv[])
{
	std::cout << "Tere" << std::endl;
	return 0;
}
```

Hello World

24.03.14

Tekst – char, std::string

Täisarvud – int, long, char, short

Ujukomaarvud – float, double

Erinevus C-ga:

Tõeväärtus - bool;

Tingimuslaused

```
if (1==2) {
        std::cout << "ei jõua siia" << std::endl;
} else if (5 < 4) {
        std::cout << "siia ka mitte" << std::endl;
} else {
        std::cout << "siin" << std::endl;
}

switch (asi) {
    case 1:
        cout << "esimene asi";
        break;
```

24.03.14

Korduslaused

```
for (int i = 0; i < 10; i++) {
	std::cout << "number " << i << std::endl;
}

int j = 0;
while (j < 10) {
	std::cout << j << std::endl;
	j += 2;
}

do {
   teeMidagi();

 } while (true);
```

24.03.14

break, continue

```
for (int i = 0; i < arv; i++ ) {
    if (i % 2 == 0)
        continue;
}

while (true) {
    int number;
    cin >> number;
    if (number > 7)
        break;
}
```

24.03.14

Operaatorid

Binaarsed ehk 2 operandiga
+, -, *, /, %
+=, -=, *=, /=

Unaarsed ehk 1 operandiga
++, --

Võrdlustehted
==, !=
<, >, <=, >=

Loogikatehted
!, &&, ||

24.03.14

Sisend

```
#include <iostream>

using namespace std;

int main (int argc, char *argv[])
{
    int number;
    cin >> number;
    for (int i = 0; i < number; i++) {
        cout << i*i << endl;
    }
}
```

24.03.14

Milleks? Koodi uuesti kasutamiseks.


```
#include <iostream>
using namespace std;

int suurenda(int n)
{
    return n + 1;
}
int main (int argc, char *argv[])
{
    int number = 2;
    number = suurenda(number);
    cout << number << endl;
}
```

Funktsioonid - Definitsioon vs deklaratioon

```
// Ei kompileeru
#include <iostream>

using namespace std;

int main (int argc, char *argv[])
{
    int number = 2;
    number = suurenda(number);
    cout << number << endl;
}

int suurenda(int n)
{
    return n + 1;
}
```

24.03.14

Funktsioonid - Definitsioon vs deklaratioon

```
// Korras!
#include <iostream>

using namespace std;
int suurenda(int); // deklaratsioon

int main (int argc, char *argv[])
{
    int number = 2;
    number = suurenda(number);
    cout << number << endl;
}
// definitsioon
int suurenda(int n)
{
    return n + 1;
}
```

24.03.14

Tavaliselt pannakse jagatud koodi deklaratsioonid ühte header faili, mida saab lisada #include abil teistesse .c/.cpp failidesse, kus nende sisu vaja läheb.

header failid (.h)

#include saab kasutada kahel viisil.

Kas
#include <failinimi.h>

või

#include "failinimi.h"

24.03.14

```
// teistmoodi suurendamine, viidaga
#include <iostream>
using namespace std;

void suurenda(int &n)
{
   suurenda++;
}
int main (int argc, char *argv[])
{
   int number = 2;
   suurenda(number);
   cout << number << endl;
}
```

```
/* teistmoodi suurendamine, viidaga
kasulik, kui vaja anda edasi suuri objekte või vaja
argumendi väärtust muuta
*/
#include <iostream>
using namespace std;

void suurenda(int &n, &k, &l)
{
   suurenda++;
}
int main (int argc, char *argv[])
{
   int number = 2, n2 = 3, n3 = 5;
   suurenda(number, n2, n3);
   cout << number << " " << n2 << " " << n3 <<
endl;
}
```

arrayd

```
#include <iostream>
using namespace std;

int main (int argc, char *argv[])
{
    for (int i = 0; i < argc, i++) {
        cout << argv[i] << endl;
    }
}
```

24.03.14

STL

```
#include <iostream>
using namespace std;

int main (int argc, char *argv)
{
    int numbrid[5]; // kindla suurusega array
    for (int i = 0; i < 5, i++) {
        cin >> numbrid[i];
   }
   double summa = 0;
   for (int i = 0; i < 5; i++) {
       summa += numbrid[i];
   }
}
```

24.03.14

STL - vector

```
#include <iostream>
#include <vector>
using namespace std;

int main (int argc, char *argv)
{
    vector<int> numbrid;
    while (true) {
        int number;
        cin >> number;
        numbrid.push_back(number);
        if (number > 100) break;
   }
   double summa = 0;
   for (vector<int>::iterator it = numbrid.begin(); it !
= numbrid.end(); it++) {
       summa += *it;
   }
    cout << summa / numbrid.size() << endl;
}
```

24.03.14

- Tõnis Märtmaa

- e-mail: martmaa@gmail.com