

# Programmeerimine



**TTÜ Robotiklubi**  
Tallinn University of Technology Robotics Club

- Mõisted
- Taustaprotsess
- C keel
  - Näiteprogramm
  - Süntaksireeglid
  - Nimed ja andmetüübid
  - Muutujate deklareerimine ja väärtuse omistamine
  - Avaldised, operaatorid
  - Funktsioonid
  - Skoop
  - Laused
  - Failid
  - Koodi kirjutamise soovitused

- **Arvutiprogramm** - kindlate reeglite järgi kirja pandud instruksioonide kogum, mis on mõeldud arvuti protsessorile täitmiseks.
- **Programmeerimiskeel** – tehiskeel arvutile instruksioonide kirja panemiseks. Keelel on kindel süntaks (lauseehitus) ja semantika (täendus). Keele abil kirja pandud lauseid kutsutakse lähtekoodiks.
- **Kompilaator** – programm, mis muudab inimesele arusaadava lähtekoodi arvutile arusaadavaks masinkoodiks. Meie kasutame AVR-GCC kompilaatorit.

# Low-level vs high level

Käskood	Operant/Operandid
<b>Masinkood</b> Binary machine language	0011011100111001 1100011001010011 1111010000010101 1111000001110010
<b>Assembler keel</b> Assembly language	ADD a,b,c MOV d, M XOR c,d
<b>Kõrgtaseme keel</b> High-level language	If n<100 then a:=b Else v[i]:=k[j-1] End;

```
#include <stdio.h>
```

```
int main(void)
```

```
{
```

```
    printf("Hello world!");
```

```
    return 0;
```

```
}
```

# Näitekood keeles C

- Programm koosneb:
  - Funktsioonidest (function) - tehted
  - Muutujatest (variable) – andmed
  - Makrodest (macro) – programmi lähtekood esitatud ühe avaldisena
- Üldist
  - Olulisim funktsioon: “main”
  - Funktsioone ja muutujaid pannakse kirja lausetena (tinglikult võib öelda, et funktsioon on liitlause, mis koosneb lihtlausest).

# C keele syntax

- Funktsiooni sisu kirjutatakse looksulgude vahele {}
- Iga lihtlause lõpeb semikooloniga ‘;’
- Valged sümbolid (tühimik):
  - Tühik – Tabulaator
  - Reavahetus
- Eralduse tüübid:
  - Kohustuslik eraldus – vähemalt 1 tühimik
  - Vabatahtlik eraldus – tühimik võib puududa
- //kommentaar

# Nimed keeles C

- *Case-sensitivity*
  - Juhan ei ole samaväärne muutuja nimi kui juHaN
- Ainult ladinatähed, numbrid ja alakriipsud
- Nimi ei või alata numbri sümboliga
- Mõned *pre-set* muutujanimed, mis on reserveeritud
  
- Nt: name, S66m, 6S6m, \_, Set\_motor\_speed



- Baas-andmetüübid
  - int, string, float...
- Massiivid
  - int[], string[]
- Struktuurid, klassid
  - struct {int, string}
  - class {int, sting}

## Andmetüübid keeles C

```
int Kapsas;  
Kapsas = 145;  
  
string Nimi = „Karl“;  
  
int array[4];  
  
struct {  
    int speed;  
    int horsepower;  
}car;  
  
car Autod[20];
```

## Muutujate deklareerimine ja väärtuste omistamine

# Tehted keeles C

- Aritmeetika:  
+ - / \* %
- Loogikaavaldised:  
! && ||
- Bitiloogika:  
~ & | ^ << >>
- Võrdlus:  
== != < > <= >=
- Omistus:  
=
- Arvutus-omistus:  
+= -= \*= /= |=

- Instruktsioonide kogum, mis täidab kindlat programmi osa ning omab omaette skoopi
- Definiitsioon:

```
int add (int a, int b)
{
    return a+b;
}
```

- Väljakutsumine:

```
int summa;
summa = add (5,9);
//summa = 5+9;
```

# Funktsioonid keeles C

- Ehk kus on mingi muutuja kättesaadav või kasutatav

```
int i = 0;
int function()
{
    int i = 3; return i;
}
int main(void)
{
    i = function();
    return 0;
}
```

*Scope*

- If-lause

```
if (a>b) {a=b;}
```

```
elseif (a<b) {b=a;}
```

```
else {a=0; b=0;}
```

- For-lause

```
int a;
```

```
for (int i=0; i<5; i++)
```

```
{
```

```
    a++;
```

```
}
```

# Laused

## Laused (2)

- While-lause

```
while (a<b)
{
    dance();
}
while(1) {}
```

- `break;` - lõpetab tsükli
- `continue;` - lõpetab hetke iteratsiooni
- `#include <stdio.h>`
- `#include "stdio.h"`

- Treppimine

# Stiil

```
class Peatus
{
    vector <Reisija*> ootajad; //private vektor
public:

    void saabusReisija(Reisija *keegi)
    {
        ootajad.push_back(keegi); //saabusReisija lükkab
    };
    void print()
    {
        for(int j=0;j<ootajad.size();j++)
            ootajad[j]->print();
    }
    void saabusBuss(Buss *mingi)
    {
        for(int i=0;i<ootajad.size();i++) //kontrollib kas
        {
            if(mingi->annaLiiniNumber()==ootajad[i]->annaLiiniNumber())
            {
                if(ootajad[i]->minePeale(mingi)) //kui ta
                {
                    ootajad.erase(ootajad.begin()+i--);
                }
            }
        }
    }
};
```

```
class Peatus
{
    vector <Reisija*> ootajad; //private vektor
public:

    void saabusReisija(Reisija *keegi)
    {
        ootajad.push_back(keegi); //saabusReisija lükkab reatusa
    };
    void print()
    {
        for(int j=0;j<ootajad.size();j++)
            ootajad[j]->print();
    }
    void saabusBuss(Buss *mingi)
    {
        for(int i=0;i<ootajad.size();i++) //kontrollib kas ootajad
        {
            if(mingi->annaLiiniNumber()==ootajad[i]->annaVajalikuLiiniNumber())
            {
                if(ootajad[i]->minePeale(mingi)) //kui ta sai bussi peale
                {
                    ootajad.erase(ootajad.begin()+i--);
                }
            }
        }
    }
};
```



## Stiil (2)

- Funktsioonide, muutujate, jms nimede formaat
  - Koodistandardid
- Kommenteerimise vajalikkus
- NB! Ühtlast stiili on lihtsam lugeda, täiendada, 3 aasta pärast uuesti aru saada



## Kompileerumine vs töötamine

30.09.2015



Raimond Vaba

---

e-mail: [raimond.vaba@robotiklubi.ee](mailto:raimond.vaba@robotiklubi.ee)

tel. +372 58874959