

Alamprogramm

- Iseseisev programmi osa, mida saab korduvalt käivitada
 - Käivitajaks võib olla peaprogramm, sel juhul sünkroonne ja etteplaneeritav tegevus
 - Käivitaja võib olla ka mingi sündmus (*event-driven*), näiteks riistvara katkestus, sel juhul asünkroonne ja peaprogrammist sõltumatu
- Andmevahetus peaprogrammiga
 - Globaalmuutujate kaudu (asünkroonse variandi puhul muud moodi ei saagi)
 - Parameetrite vahendusel
- Alamprogramm võib omakorda käivitada teisi alamprogramme või ka uuesti rekursiivselt iseennast
- Reeglina on alamprogrammil unikaalne nimi / identifikaator, mille abil teda välja kutsutakse

Miks alamprogramme üldse vaja (eelised)

- Võimaldavad suurt ülesannet tükeldada väiksemateks osadeks
 - Väikesest koodijupist lihtsam omada ülevaadet
 - Lihtsam testida
- Vähendavad vajadust kirjutada sama asja mitu korda, muudavad koodi lühemaks
 - Kui vaja midagi muuta, siis tuleb seda muuta ainult ühes kohas
- Kui alamprogrammid asuvad eraldi moodulis, siis saab neid mõnes teises projektis uuesti kasutada
- Lihtsam projekti mitme arendaja vahel jagada

Puudused

- Alamprogrammi käivitamisega kaasneb teatav lisatöö
 - Võtab *stack* mälu
 - Rekursiivne alamprogramm võib programmi kokku jooksutada
- "Hea" alamprogramm on abstraktne, konkreetse ülesande lahendamiseks vaja luua palju abstraktset funktsionaalsust, mida võibolla enam kellelegi kunagi vaja ei lähe
- Ülesande tükeldamisega võib liiale minna
 - Näiteks *get* ja *set* meetodid objektorienteeritud programmeerimises

Alamprogrammid C-s

- C-s on kõik alamprogrammid funktsioonid

- Deklareerimine:

```
tüüp funktsiooni_nimi(tüüp parameeter,...) {  
    sisu  
    return väärtus;  
}
```

- Kui funktsioon midagi ei tagasta on ta tüübiks *void* ja return käsk on ilma väärtuseta või võib üldse ära jätta
- Kui funktsioonil pole parameetreid, siis sulgudes ka *void*

```
void funktsiooni_nimi(void) { sisu }
```

- *return* katkestab alamprogrammi täitmise
- Funktsiooni väljakutsumisel sulud vajalikud ka siis kui parameetreid pole

Globaalmuutujad

- Kui muutujad on defineeritud väljaspool alamprogramme, siis nad on kättesaadavad kõikidele alamprogrammidele, st suvaline funktsioon saab neid lugeda ja muuta.*
- Andmevahetus eri alamprogrammide vahel on peidetud, raske on omada ülevaadet mis muutujaid üks või teine funktsioon kasutab, seetõttu on keeruline üles leida vigu.

```
int x; //globaalmuutuja
void loeAndmed(void) {
    printf("Sisesta arv: ");
    scanf("%d", &x); //seesama globaalmuutuja
}
void kuvaAndmed(void) {
    printf("%d\n", x); //seesama globaalmuutuja
}
```

Lokaalmuutujad

- Lokaalmuutujad on defineeritud alamprogrammi sees ja on nähtavad ainult selles alamprogrammis.
- Mõnes muus alamprogrammis võib olla sama nimega muutuja, kuid nad on erinevad ja ühe muutmine ei muuda teist.

```
void loeAndmed(void) {
    int x; //lokaalmuutuja
    printf("Sisesta arv: ");
    scanf("%d", &x); //sama lokaalmuutuja, mis 2 rida enne
}

void kuvaAndmed(void) {
    int x; //uus lokaalmuutuja, pole sama, mis enne
    printf("%d\n", x); //sama, mis rida kõrgemal
}
```

Parameetrid ja return käsk

- Kui on soov lokaal muutujate või mingi konstandi väärtust edastada alamprogrammist välja, siis selle jaoks on return käsk (sellele, kes meid välja kutsus) või parameetrid (sellele, keda me välja kutsume).
- Kui tegemist pole viidaparameetriga, siis edastatakse ainult väärtust. Kui parameetriga edastatud väärtust alamprogrammis muudetakse, siis peaprogrammis see ei muutu.

```
int main(void) {  
    int k = 5; //peaprogrammi muutuja  
    arvuta(k); //kutsutakse välja alamprogramm arvuta  
}  
  
void arvuta(int k) { //parameeter  
    k = k + 2; // kohalik k väärtus muutub, peaprogrammi oma mitte  
}
```

Parameetrid ja return käsk

```
int loeAndmed(void) {
    int x; //lokaalmuutuja
    printf("Sisesta arv: ");
    scanf("%d", &x); //sama lokaalmuutuja, mis 2 rida kõrgemal
    return x; //väärtus edastatakse kutsujale
}

void kuvaAndmed(int x) { //parameeter, pole eelnevaga sama
    printf("%d\n", x); //trükitakse välja parameetri väärtus
}

int main(void) {
    int x; //peaprogrammi lokaalmuutuja, pole sama, mis eelnevad
    x = loeAndmed(); //saadud väärtus omistatakse x-le.
    kuvaAndmed(x); //x väärtus antakse parameetrina edasi
}
```


Programmi voog

- Peaprogrammi main() oleme me katkestanud return käsuga. Näiteks kontrollime, et kasutaja sisestatud parameetrid ei sobi ja väljume kogu programmist.
- Alamprogrammist väljudes satutakse selliselt tagasi peaprogrammi. Kui oleks vaja väljuda rakendusest keset alamprogrammi, siis on abiks funktsioon exit(), mis on defineeritud stdlib.h teegis.
- Kui meil on vaja kontrollida mitut asja, siis peaprogrammis peaksime selle jaoks ehitama mitu if – else plokki. Kui alamprogrammis kontrollile järgneb return käsk, siis saab else laused ära jätta

```
if (a < b)
```

```
    return 5;
```

```
// siia satutakse ainult siis, kui a >= b
```

```
if (b < c)
```

```
    return -5;
```