Struktur - biblioteker (libraries)

- Struktur
- Design af “generaliseret” kode
- i C

- Interface til bibliotek (.h fil)
- Selve koden (.c fil) (I Arduino .cpp)

- eksempel

eks 1: average

- Interface
  - Datastructure

```
#define VERSION "1.23A"

struct averTp {
  int arraySize;
  int arrayIndex;
  int arrayCount;
  int *ptr2array;
};
```

```
const int arSz = 50;
struct averTp myData;
int averAr[arSz];
```

eks 1: average

- Interface
  - Datastructure

```
#define VERSION "1.23A"

struct averTp {
  int arraySize;
  int arrayIndex;
  int arrayCount;
```

```
const int arSz = 50;
struct averTp myData;
int averAr[arSz];

//   initAverArData(&myData, averAr, arSz);  //(  myData  == &(myData[0] )
```

```
const int arSz2 = 23;
int ar2[arSz2];
struct averTp myData2 = { .arraySize = arSz2, .arrayIndex = 0, .arrayCount = 0, .ptr2array = ar2};
```

```
const int arSz = 50;
struct averTp myData;
int averAr[arSz];
```

```
int initAverArData(averTp *averInfo, int *averAr, int averArSize) {
  averInfo->arraySize = averArSize;
  averInfo->arrayIndex = 0;  // init
  averInfo->arrayCount = 0;
  averInfo->ptr2array = averAr;
}
```

```
initAverArData(&myData, averAr, arSz);  //(  myData  == &(myData[0] )
```

add a new value

```
int addValue(averTp *averInfo, int newVal) {

  averInfo->ptr2array[averInfo->arrayIndex] = newVal;

  if (averInfo->arrayIndex < averInfo->arraySize - 1) {
    (averInfo->arrayIndex)++;
  } else {
    averInfo->arrayIndex = 0;
  }

  if (averInfo->arrayCount < averInfo->arraySize) {
    averInfo->arrayCount++;
  }
  return 0;  // ok
}
```

calc average

```c
int calcAver(averTp *averInfo) {
  int sum = 0;
  for (int i = 0; i < averInfo->arrayCount; i++) {
    sum += averInfo->ptr2array[i];
  }
  return sum / averInfo->arrayCount;
}
```

Interface

- Data struct
- Functions:
-

```
#define VERSION "1.23A"

struct averTp {
  int arraySize;
  int arrayIndex;
  int arrayCount;
  int *ptr2array;
};
```

```
int initAverArData(averTp *averInfo, int *averAr, int averArSize)

int addValue(averTp *averInfo, int newVal)

int calcAver(averTp *averInfo)
```

bibliotek - Arduino way

- Folder i libraries med biblioteksnavn: averageint
- kildetekst i src folder
  - header fil (.h) med samme navn som folder: averageint.h
  - kildetekst i (.cpp) med samme navn som folder: averageint.cpp
    - (std C “kræver .c” Arduino kræver c++ (c plusplus)
- eksempelkode i examples folder

averageint.h

```
// averageint.h

#ifndef AVERAGEINT

#define AVERAGEINT

#define VERSION "1.23A"

struct averTp {
  int arraySize;
  int arrayIndex;
  int arrayCount;
  int *ptr2array;
};

// Function interface afsluttet med ;

int initAverArData(averTp *averInfo, int *averAr, int averArSize);
int addValue(averTp *averInfo, int newVal);
int calcAver(averTp *averInfo);


#endif
```

Arduino weirdness - igen

- kræver kildetekst fil hedder .cpp (c plusplus og ikke c som efternavn)

averageint.cpp (1/2)

```
int initAverArData(averTp *averInfo, int *averAr, int averArSize) {
  averInfo->arraySize = averArSize;
  averInfo->arrayIndex = 0;  // init
  averInfo->arrayCount = 0;
  averInfo->ptr2array = averAr;
}

int calcAver(averTp *averInfo) {
  int sum = 0;
  for (int i = 0; i < averInfo->arrayCount; i++) {
    sum += averInfo->ptr2array[i];
  }
  return sum / averInfo->arrayCount;
}
```

averageint.cpp (2/2)

```
int addValue(averTp *averInfo, int newVal) {
  averInfo->ptr2array[averInfo->arrayIndex] = newVal;

  if (averInfo->arrayIndex < averInfo->arraySize - 1) {
    (averInfo->arrayIndex)++;
  } else {
    averInfo->arrayIndex = 0;
  }

  if (averInfo->arrayCount < averInfo->arraySize) {
    averInfo->arrayCount++;
  }
  return 0;  // ok
}
```

```
libraries -->  averageint
                  -->  src
                           -->  averageint.h
                           -->  averageint.cpp
                  --> examples
                           --> ex01
                                  -->  ex01.ino
```

Folder zippes og kan installeres som bibliotek fra Ardino

https://jensd.dk/doc/arduino/libraries/

Hærde et program - skjul det der skal skjules

- Man kan skjule variable og funktioner for resten af et program
- Det er på filniveau dvs

- Man kan skjule variable og funktioner så de kun kan ses/bruges/kaldes i den fil (.c) de er erklæret i.

static som i “henmmelig”

main.c
links sammen med lib1.c

```
extern int pubInt;
void publicFct();   // prototype ligesom i .h
extern static int hemmeligVar;  // duer ikk


void main()
{

  publicFct();        // ok
  pubInt = 33;        // ok
  hemmeligFct();      // not ok
  hemmeligVar = 345;  // not ok
}
```

lib1.c

```
// public variabel og function

// skjulte i denne fil

static int hemmeligVar;

static void hemmeligFct()
{
}


int pubInt;
void publicFct()
{
    hemmeligFct();  // ok
    hemmeligVar = 3; (( ok
}
```

```
// hovedprogram


extern int ii;

extern int gg;

void setup() {

  ii = 3;

  hemmelig = 7;  // kan compiler ikke finde

   // put your setup code here, to run once

}


void loop() {

}
```

```
// en Tab i hovedprogram


int ii;

static int hemmelig;   // hemmelig
```

hvorfor

- For at undgå folk laver dumme utilsigtede ting.

skjule definition af struct

```
// WHAT USER CAN SEE include this file (.h file)
Integer.h

#pragma once

#include <stdio.h>

typedef struct Integer Integer;

/*
 * The compiler considers struct Integer
 * as an incomplete type.
 * Incomplete types can be used in declarations.
 */

size_t integerSize();

void integerSetValue(Integer* i, int val);

int integerGetValue(Integer* i);
```

```
// C source  .C file

#include "Integer.h"  // for validity check

typedef struct Integer {  // Here  you define your struct
    int value;
} Integer;

/*
 * The caller will handle allocation.
 * Provide the required information only
 *
 * only access to struct by function call and parameters
*/

size_t integerSize() {
    return sizeof(Integer);
}

void integerSetValue(Integer* i, int val) {
    i->value = val;
}

int integerGetValue(Integer* i) {
    return i->value;
}
```

I

```
#include <Integer.h>

Integer minVariabel;
int minV;

IntegerSetValue(&minVariabel, 3);

minV = IntegerGetValue(&minVariabel);


JEG KAN IKK

minVariabel.value = 333; // !!!!
```