Øvingsforelesning 4

TDT4102 — Prosedyre- og objektorientert programmering

Iver Karlsbakk Småge, *und.ass.*
Torje Nysæther, *und.ass.*

3. februar, 2021

IDI, NTNU

Plan for dagen

Pass-by-value vs. pass-by-reference

Struct

Char/string

Pass-by-value vs. pass-by-reference

Pass by value

```
int increment(int number) {
    number += 1;
    return number;
}

//
int num = 5;
int incNum = increment(num);
cout << num; //5
//
```

Pass by reference

```
int increment(int & number) {
    number += 1;
    return number;
}

//
int num = 5;
int incNum = increment(num);
cout << num; //6
//
```

Pass by value/reference

```
int foo(int & n1, int n2) {
    n1 += n1*n2;
    return n1+n2
}

//
int n1 = 2;
int n2 = 1;
int n3 = foo(n1, n2);
//Hva er n1, n2 og n3?
//
```

Pass by value/reference

```
int foo(int & n1, int n2) {
    n1 += n1*n2;
    return n1+n2
}

//
int n1 = 2;
int n2 = 1;
int n3 = foo(n1, n2);
//Hva er n1, n2 og n3?
// n1 = 4, n2 = 1, n3 = 5
//
```

Pass by reference

- Gjør at vi ikke trenger å returnere noe

```
void increment(int & number) {
    number += 1;
}

//
int num = 5;
increment(num);
cout << num; //6
//
```

Pass by const reference

- Brukes når vi ikke skal endre på verdien

```
int increment(const int & number) {
    number += 1; //Not allowed
    return number;
}

int increment(const int & number) {
    int number_2 = number + 1; //Allowed
    return number_2;
}
```

Spørsmål?

Plan for dagen

Pass-by-value vs. pass-by-reference

Struct

Char/string

Struct

Typer i C++

- Alle variabler i C++ har en type.
- Fundamentale datatyper er for eksempel `int`, `bool`, `double`, `char`
- Mer kompliserte datatyper er for eksempel `string`, `vector` og `AnimationWindow`
- Hver type har et sett med operasjoner som kan utføres på dem. For eksempel har:
  - `vector` metoden `push_back()`
  - `AnimationWindow` metoden `draw_triangle()`

Egendefinerte typer - Struct

- Vi ønsker å lage våre egne typer
- Dette kan gjøres ved å sette sammen andre typer
- På denne måten kan vi organisere dataen vår bedre
- For eksempel `AnimationWindow` er en egendefinert type, laget av Bart (foreleser)
- Egendefinerte typer er starten på objektorientert programmering - men mer om dette i øving 5

Hva er et struct?

Istedenfor å lage fire variabler kan vi lage én struct.

```
struct WeatherData{
    string location;
    int temperature;
    int windStrength;
    int rainfall_mm;
};
```

Verdiene inni structet kalles medlemsvariabler!

Initialisering av struct

```
struct Fridge{
    string brand;
    int numApples;
    int numCarrots;
};
```

- ```
  //Med krøllparantes
  Fridge f{"Bosh", 3, 5};
  ```
- ```
  //Med dot-operator
  Fridge f{};
  f.brand = "Samsung";
  f.numApples = 2;
  f.numCarrots = 2;
  //
  ```
- Hva skjer om man ikke initialiserer et struct?

Dot-operatoren

- Vi bruker dot-operatoren både til å hente ut medlemsvariabler fra structet, og skrive til medlemsvariabler i structet.

```
Fridge f{"Logik", 2, 2};
string name = f.brand;
f.numApples = 3;

//Vi kan gjøre alle operasjoner vi kan gjøre på medlemsvariablene.
//Her blir 'f.numCarrots' 6
f.numApples += f.numApples + 2;
```

Structs fungerer som andre typer

```
//Kan brukes i f.eks. vectorer
vector<Fridge> fridges;
fridges.push_back(Fridge{"Siemens",1,2});

 //Brukes som vanlig i funksjoner
Fridge increaseCarrots(Fridge f) {
    f.carrots++;
    return f;
}
```

Livekoding

En liten sneak-peak

- Vi har bare gått gjennom noe av funksjonaliteten til struct. Neste øving handler om klasser, som er nesten det samme. Mye av det dere lærer om klasser gjelder også for structs!

Spørsmål?

Plan for dagen

Pass-by-value vs. pass-by-reference

Struct

Char/string

Char/string

Char

- En char er i C++ representert som et tall mellom 0 og 255.
- Det oversettes til tegn ved hjelp av ASCII-tabellen.

Char-aritmetikk

- Dette gjør at vi kan finne på litt fanteri.

```
cout << static_cast<char>(65); // A
```

Nyttig å vite om char-aritmetikk

- `'a' < 'z' // True`
- `'a' < 'Z' // False` Store bokstaver er mindre enn små.
- `'A'-1` er @
- `'a'-1` er ‘ (en backtick)
- `'Z'+1` er [
- `'z'+1` er {
- Hvis dere ser noen av disse (sannsynligvis @) i oppgaven der dere skal generere en random string, er det sannsynligvis fordi dere har gjort noe feil i char-aritmetikken.

String

- En string er en liste med bokstaver.

String

- En string er en liste med bokstaver.
- ...Med litt ekstra magi
- Hvorfor bruker vi `string` over disse datatypene?
- `char[]`
- `vector<char>`

Livekoding

Livekoding (hvis vi har tid).

Spørsmål?