

Imperativ Programmering og Datastrukturer

Datastrukturer: records

René Rydhof Hansen

19. november 2008

(opdateret 24. november 2008)

Mål

- At kunne definere hvad en datastruktur er
- At kunne definere og bruge records (sammensat type)

Definition

Data sammensat af forskellige datatyper, relationer mellem dem og operationer på dem.

- Organisering af data
- Effektivitet

- Arrays
- Records
- Linked lists (hægtede lister)
- Stakke
- Køer
- (Træer)
- ((Grafer))

Hvorfor Records (sammensatte typer)?

Arrays

```
String titel [117];  
int     udg_aar [117];  
int     spilletid [117];
```

- Spredt information
- Vanskeligt at se sammenhænge
- Ændringer skal ske mange steder

Definition af "Records"

"Records"

```
class Album
{
    String titel;
    int    udg_aat;
    int    spilletid;
}
```

- Grupperer/samler et antal variable, muligvis af forskellig type, til et samlet logisk hele.
- De enkelte variable kaldes *felter* (fields)
- Hvert felt har et navn
- Felter tilgås med det *kvalificerede navn* (dot notation):
record.fieldname

Definition af "Records"

"Records" ... i praksis

```
public class Album // placeres efter Form
{
    public String titel;
    public int    udg_aat;
    public int    spilletid;
}
```

- Grupperer/samler et antal variable, muligvis af forskellig type, til et samlet logisk hele.
- De enkelte variable kaldes *felter* (fields)
- Hvert felt har et navn
- Felter tilgås med det *kvalificerede navn* (dot notation):
record.fieldname

Example

```
Album alb = new Album();  
  
alb.titel = "FooBarBaz";  
alb.udg_aar = 1963;  
alb.spilletid = 87;  
  
MessageBox.Show("Titel:" + alb.titel);
```

Example (...nu med færre fejl)

```
Album[]  albs = new Album[100];  
  
albs[42] = new Album();  
  
albs[42].titel = "FooBarBaz";  
albs[42].udg_aar = 1963;  
albs[42].spilletid = 87;
```

Example (Samlet Spilletid)

```
for(int i = 0, sum = 0; i < 100; i++) {  
    sum += albs[i].spilletid;  
}
```