Objekter

INF101 forelesning 24. Januar 2022

Torstein Strømme

Stikkord: Klasser, objekter, primitive og refererte typer, minne, static vs ikke-static

Returnere flere verdier

```
def return_winner():
    the_score = 0
    the_player = "Player A"
    return the_player, the_score
```

Returnere flere verdier

```
ScoreAndPlayer returnWinner() {
    int theScore = 0;
    String thePlayer = "Player A";
    return new ScoreAndPlayer(theScore, thePlayer);
}

class ScoreAndPlayer {
    int score;
    String player;

    ScoreAndPlayer(int score, String player) {
        this.score = score;
        this.player = player;
    }
}
```

Et objekt

- Har *feltvariabler*

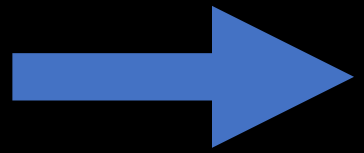
```
public static void main(String[] args) {
    ScoreAndPlayer winner = new ScoreAndPlayer(0, "Player A");
    System.out.printf("Player %s won with %d points\n", winner.player, winner.score);
}
```

Et objekt

- Har *feltvariabler*
- Kan utføre *metoder*

```
public static void main(String[] args) {
    ScoreAndPlayer winner = new ScoreAndPlayer(0, "Player A");
    winner.increaseScore(10);
    System.out.printf("Player %s won with %d points\n", winner.player, winner.score);
}
```

Et objekt



er
(en instans av)

En klasse

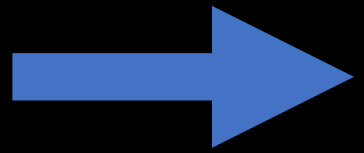
- Har *feltvariabler*
- Kan utføre *metoder*

- Beskriver *typen* feltvariabler
- Beskriver oppførselen til metoder

Objekt	Klasse
Per, f. nr: 240122 32123	Person
DE83556	Bil
42 + 5i	Komplekst tall
0xDEADBEEF	Minneadresse

Et objekt

- Har *feltvariabler*
- Kan utføre *metoder*



er
(en instans av)

En klasse

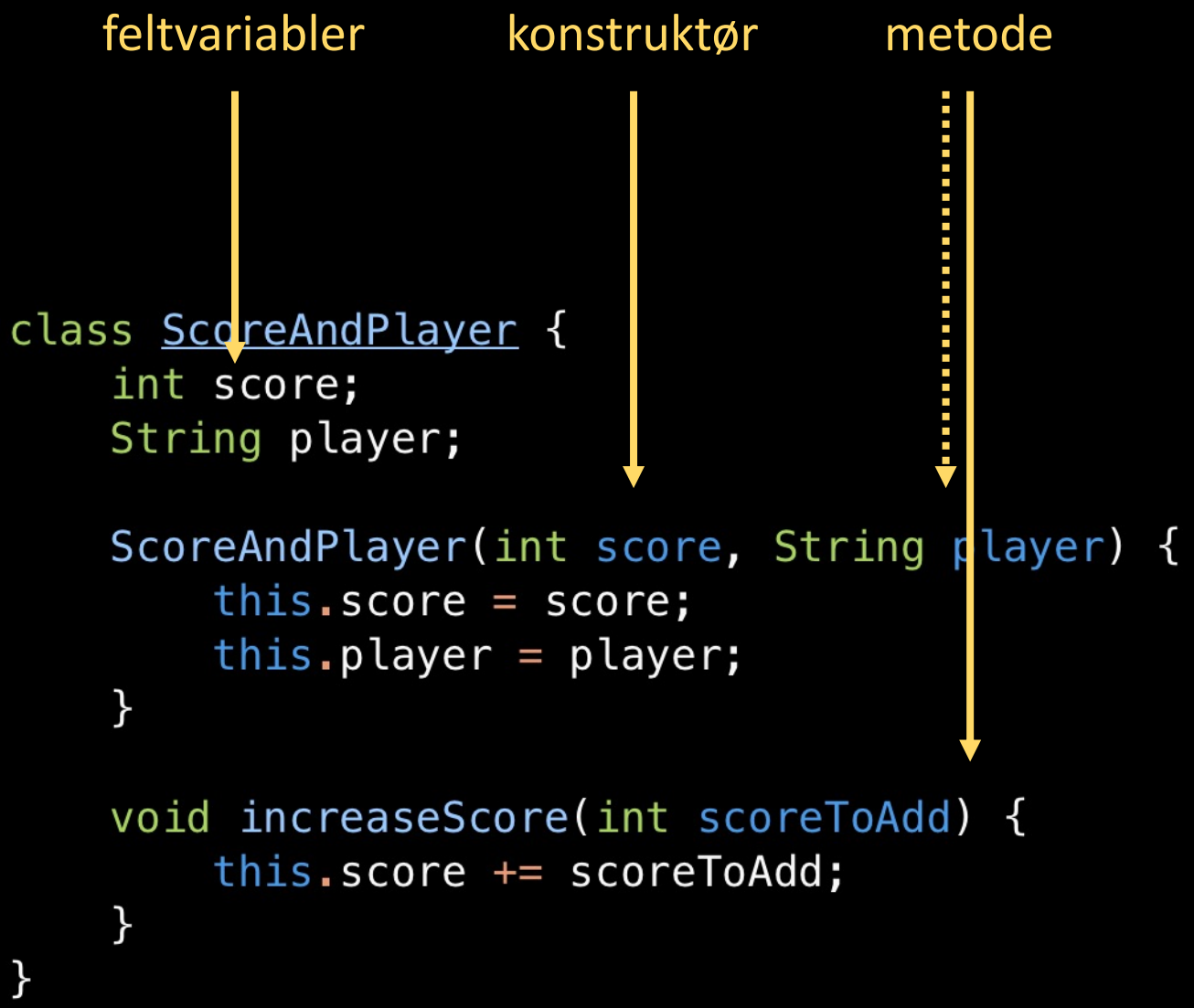
- Beskriver *typen* feltvariabler
- Beskriver oppførselen til metoder

```
public static void main(String[] args) {
    ScoreAndPlayer winner = new ScoreAndPlayer(0, "Player A");
    winner.increaseScore(10);
    System.out.printf("Player %s won with %d points\n", winner.p
}
```

```
class ScoreAndPlayer {
    int score;
    String player;

    ScoreAndPlayer(int score, String player) {
        this.score = score;
        this.player = player;
    }

    void increaseScore(int scoreToAdd) {
        this.score += scoreToAdd;
    }
}
```



En klasse

- Beskriver *typen* feltvariabler
- Beskriver oppførselen til metoder
- Konstruktør-metoder oppretter nye objekter
  - Konstruktører har kobinert navn og returverdi

Hvorfor statiske typer?

- Finn feilen så fort som mulig

- Selv-dokumenterende kode
- Enklere å feilsøke
- Du kan enkelt finne ut *hvor* en klasse er brukt
- Krever at andre som bruker koden din, bruker den med riktig input
- Forklarer til andre som ser på/bruker koden din hva du gir som output
- Du skriver 15% mindre bugs i typede språk

https://ieeexplore.ieee.org/document/7985711

Eksempel: FireProtectionSystem

- Lag et system for brannvarsling
- Det skal være mulighet for flere alarmer og flere sensorer
- Alle alarmer skal være på så lenge minst én sensor er aktiv

- Prinsipp for objektorientert programmering:
- En klasse er en abstraksjon av en (fysisk eller logisk) enhet

”Single-responsibility principle” ”do one thing, and do it well”

Eksempel: FireProtectionSystem