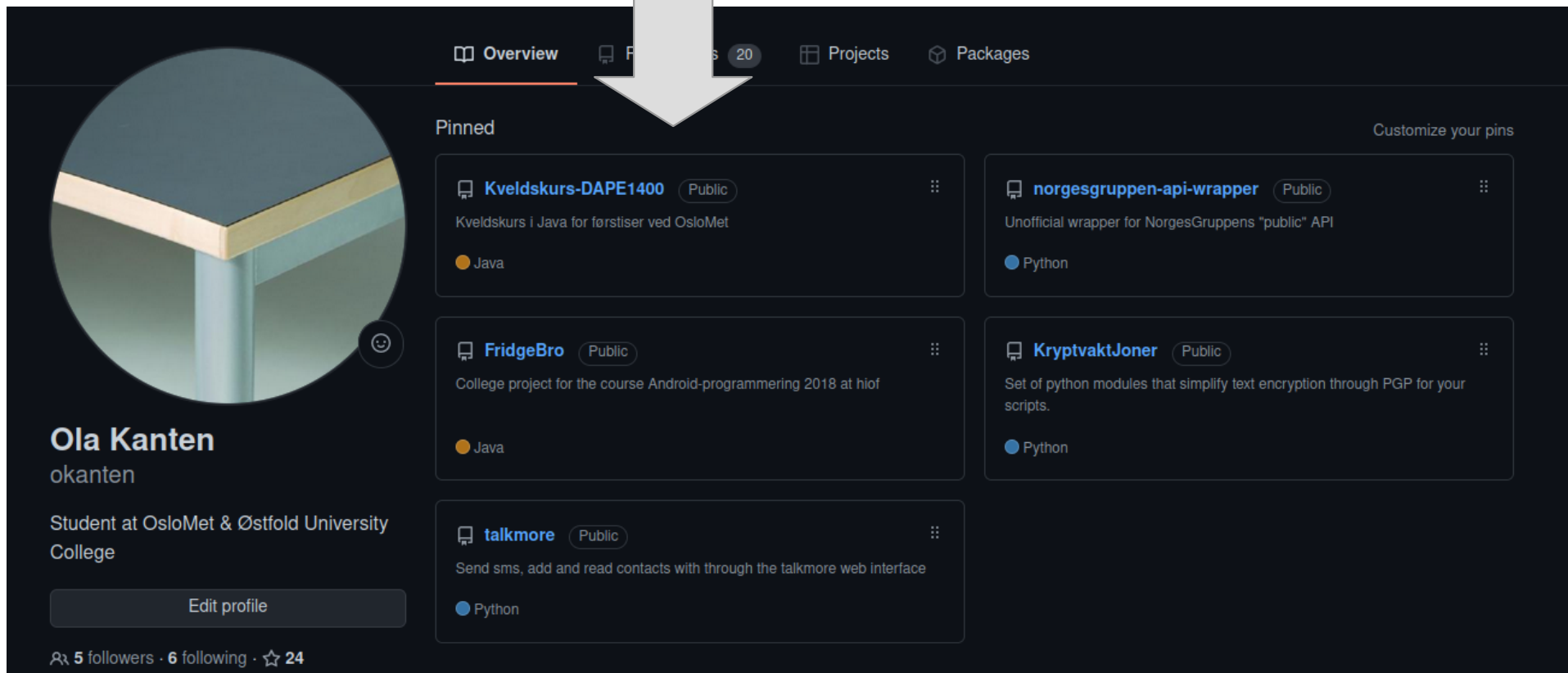


**Kveldskurs DAPE1400**

# GitHub

- Slides og kode blir lagt ut på github
- <https://github.com/okanten>



The screenshot shows a GitHub profile page for **Ola Kanten** (username: okanten). The profile is a student at OsloMet & Østfold University College, with 5 followers, 6 following, and 24 stars. The 'Pinned' section displays five projects:

- Kveldskurs-DAPE1400** (Public): Kveldskurs i Java for førstiser ved OsloMet. Language: Java.
- norgesgruppen-api-wrapper** (Public): Unofficial wrapper for NorgesGruppens "public" API. Language: Python.
- FridgeBro** (Public): College project for the course Android-programmering 2018 at hiof. Language: Java.
- KryptvaktJoner** (Public): Set of python modules that simplify text encryption through PGP for your scripts. Language: Python.
- talkmore** (Public): Send sms, add and read contacts with through the talkmore web interface. Language: Python.

A large white arrow points to the 'Pinned' section header.

# Eksamenstips

- Jeg ble lært til å bruke ordet «feltvariabler» istedenfor «attributter»
  - Bruk ordet «attributter» på eksamen
- Gjør eksamen lesbar og lett å rette
  - Følg konvensjoner dere har lært i kurset
  - Vis deg litt frem, men ikke for mye
    - Ikke gå utenfor det oppgaven ber om

```
1  var $ = function(id) {  
2    |   return document.getElementById(id);  
3  }  
4  
5  window.addEventListener("load", function() {  
6    |   var counter = 0;  
7    |   $("btnCount").addEventListener("load", function() {  
8    |       |   counter++;  
9    |       |   $("txtCounter").value = counter;  
10   |   })  
11  })  
12
```

```
16  window.onload = oppstart;
17
18  function oppstart() {
19      document.getElementById("btnCount").onclick=tellOpp
20  }
21
22  function tellOpp() {
23      counter++;
24      document.getElementById("txtCounter").value = counter;
25  }
26
```

# Arv (inheritence)

- Som foreldre og barn
  - Parent class, child classes
- En klasse kan kun ha en parent
- Foreldre kan ha mange barn
- Alle barn har bare én forelder

# Mer om arv

- Barn arver fra parent class
  - Akkurat som mennesker arver gener fra foreldre
- Klassen arver attributter (verdier) og metoder (kunnskap)
  - Akkurat som mennesker kan arve penger og kunnskap fra foreldre
- Barn av en bilmekaniker kan trolig mye om bil
  - Klassen tilhørende bilmekaniker har metoder for å fikse bil




```
public class Person {  
    private String navn;  
    private int alder;  
    private long formue;  
    private Yrke yrke;  
    private double formue0kningsprosent = 5;  
  
    public Person(String navn, int alder, long formue, Yrke yrke) {  
        this.navn = navn;  
        this.alder = alder;  
        this.formue = formue;  
        this.yrke = yrke;  
    }  
}
```

```
public class Barn extends Person {  
    private Person forelder;  
    public Barn(String navn, int alder, Person forelder) {  
        super(navn, alder, formue: 0, yrke: null);  
        this.forelder = forelder;  
        overforFormue();  
    }  
  
    public void overforFormue() {  
        this.setFormue((long) (forelder.getFormue() * 0.10));  
        forelder.setFormue(forelder.getFormue() - this.getFormue());  
    }  
}
```

# this

- Kan være vanskelig å forstå
  - Er som å sykle
- Refererer til denne instansen av et objekt
  - `person1.getNavn()`
    - `return this.navn;`

 Person.java

```
1  public class Person {
2      private String navn;
3      private int alder;
4
5      public Person(String navnFraInitialisering, int alderFraInitialisering) {
6          this.navn = navnFraInitialisering;
7          this.alder = alderFraInitialisering;
8      }
9
10     public String getNavn() {
11         return this.navn;
12     }
13
14     public void setNavn(String nyttNavn) {
15         this.navn = nyttNavn;
16     }
17 }
```

# Access Modifiers

- Brukes for attributter, klasser og metoder
- Disse er de mest relevante:
  - public
  - private
  - protected

# public

- *Kan* brukes på attributter, klasser og metoder
- Betyr ikke at du skal
  - Mest brukt til klasser og metoder
  - Sjeldent brukt på attributter
    - Med mindre de er final eller static, eller begge.

# private

- *Kan* brukes på attributter og metoder
- Betyr ikke at du skal
  - Brukes på attributter
  - Brukes på metoder i noen tilfeller
    - Ofte når metoden brukes internt i klassen

# protected

- *Kan* brukes på attributter og metoder
- Brukes med arv (inheritence)
  - Mellomting mellom private og public
  - Metoder og attributter blir tilgjengelig i child classes, men ingen andre steder



# Abstract class

- Klasse som ikke kan instansieres (lages objekter av)
  - Brukt som parent class
- Parent class *må* ikke å være abstract