IN1010 - Seminar 2

- python → Java
  - I/O

Praktisk

Oblig 1 er nå lagt ut, frist 1. februar kl 23.59

Husk å sjekke emnesiden regelmessig

Møt opp i forelesning! Ikke alltid opptak er tilgjengelig før gruppetimen

Vise hvordan man melder seg på lab

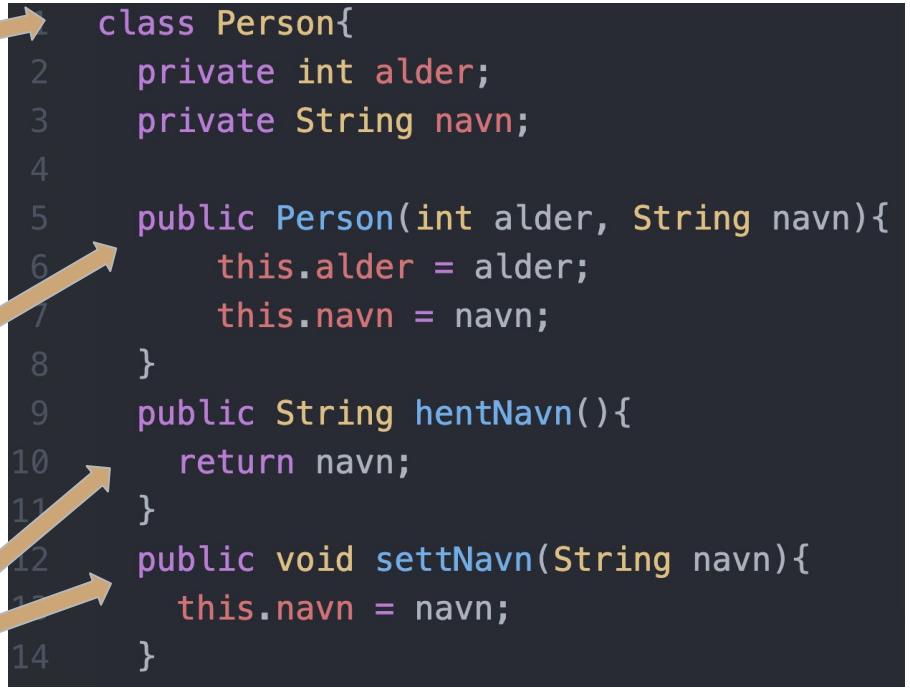
Vise fram notat med datastrukturtegning-syntaks!

https://www.uio.no/studier/emner/matnat/ifi/IN1010/v21/index.html

Repetisjon forrige uke

Private/Public

- Hvis man ikke skriver private foran class er den public
- Alt som er public kan aksesseres utenfor klassen
- Konstruktøren må være public for at man skal kunne lage instanser av klassen utenfor klassen
- God skikk at variablene er private, og at man endrer og henter dem ved hjelp av metoder (hent og sett)



```
class Person{
  private int alder;
  private String navn;

  public Person(int alder, String navn){
      this.alder = alder;
      this.navn = navn;
  }
  public String hentNavn(){
    return navn;
  }
  public void settNavn(String navn){
    this.navn = navn;
  }
```

Static

- Lager en klasse Fjellrev med en static variabel rodlistet
- Navn og alder er ikke static.
- Hvis en art er rødlistet gjelder det for alle individer av arten: klassevariabel
  - Derfor er rødlistet static, den gjelder for alle instanser av klassen!
- Men de har ulike navn og ulike alder: instansvariabler
  - Derfor er navn og alder ikke static, de er ikke det samme for alle instanser!
- Oppretter to fjellrever
- Endrer fjellrev1.rodlistet til false, da blir også fjellrev2.rodlistet false!

```java
class Fjellrev{
  private static Boolean rodlistet = true;
  private String navn;
  private int alder;

  public Fjellrev(String navn, int alder){
    this.navn = navn;
    this.alder = alder;
  }
  public void setRodlistet(Boolean rodlistet){
    this.rodlistet = rodlistet;
  }
  public Boolean hentRodlistet(){
    return rodlistet;
  }
}
```

```java
class TestFjellrev{
  public static void main(String[] args) {
    Fjellrev fjellrev1 = new Fjellrev("Bjarne", 5);
    Fjellrev fjellrev2 = new Fjellrev("Alfred", 10);
    fjellrev1.setRodlistet(false);
    System.out.println(fjellrev2.hentRodlistet());
  }
}
```

Output: >> false

Returtype (f.eks. void)

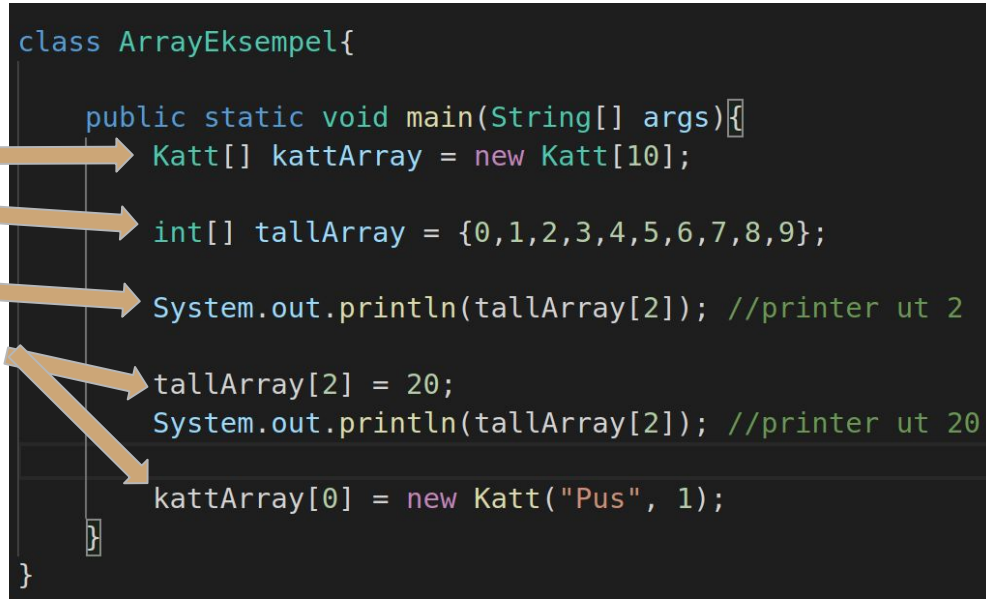
- I java må hver metode ha en returtype!
- Void betyr at metoden ikke returnerer noe
- Alle datatyper og klasser kan være returtype
- Her er eksempler med Boolean, String, int og klassen Fjellrev
- Returtype matcher typen til det vi skal returnere!
  - f.eks. bestevenn er et objekt av klassen Fjellrev, se linje 5 i koden (der bestevenn er deklarert)!

```
1  class Fjellrev{
2    private static Boolean rodlistet = true;
3    private String navn;
4    private int alder;
5    private Fjellrev bestevenn;
6
7    public Fjellrev(String navn, int alder){
8      this.navn = navn;
9      this.alder = alder;
10   }
11   public void setRodlistet(Boolean rodlistet){
12     this.rodlistet = rodlistet;
13   }
14   public Boolean hentRodlistet(){
15     return rodlistet;
16   }
17   public String hentNavn(){
18     return navn;
19   }
20   public int hentAlder(){
21     return alder;
22   }
23   public Fjellrev hentBestevenn(){
24     return bestevenn;
25   }
26 }
```

Array

- Kan legge elementer av samme type etter hverandre i minne
  - Også objekter
- Man kan opprett arrays på to måter
  - Lage et nytt “tomt” array
  - Lage array med elementer
- Man kan hente ut et element i Array ved bruk av indeksen
- Det er også slik man kan endre verdien
- Du kan også lage et array av array (det vi i IN1000 kalte nøstet liste)

NB: Ingen klasse, har ingen metoder



Send Johanna en direktemelding i chatten

1. Vil du jobbe med samme gruppe som forrige uke? Da trenger du bare svare på dette spørsmålet!
2. Er du komfortabel med å gjøre noen oppgaver med andre på zoom?
3. Hvor godt forstår du stoffet fra denne uken på en skala fra 1(lite godt) - 6(veldig godt)?
4. Noen spesielle du vil jobbe med?

Hva er forskjellen på lister(python) og arrays(Java)?

Skriv ned på ark/pc

Hva er likhetene til lister(python) og arrays(java)?

Skriv ned på ark/pc

Diskuter sammen i breakout rooms

lister(python) VS. arrays(Java)

Likheter:

- Kan itereres over
- Tar vare på flere enn en verdi
- Sortert
- Alle verdier har en indeks
- Kan ha flere forekomster av samme verdi (i motsetning til mengde)

Ulikheter:

- dynamisk VS statisk
  - Arrays har en fast lengde, den kan ikke bli endret
- Forskjellige typer VS kun en type
- Arrays i Java er mer effektive
- Lister i python har metoder, det har ikke arrays i java
- len(pythonList) vs javaArray.length

Repetisjon denne uken

ArrayList

- Er en klasse og har metoder
  - Må importers: java.util.ArrayList
  - Link til dokummentasjon: https://docs.oracle.com/javase/8/docs/api/java/util/ArrayList.html
- Er dynamisk
  - Du kan dermed legg til og fjerne elementer.
- Også kun en type, du må fortelle hvilken
- Greie metoder å kunne: (E er typen du har valgt)
  - add(E e) : legge til et element
  - size(): returner størrelsen på arraylisten
  - get(int index): henter et element på en gitt indeks
  - set(E e, int index): setter en verdi på en gitt indeks
  - remove(int index): fjerne et element på en gitt index
  - isEmpty(): sjekker om lista er tom (boolean)
  - clone(): returnerer en kopi av listen
  - clear(): fjerner alle elementene fra lista
  - toArray(): konverter ArrayListen til ett array og returnerer det

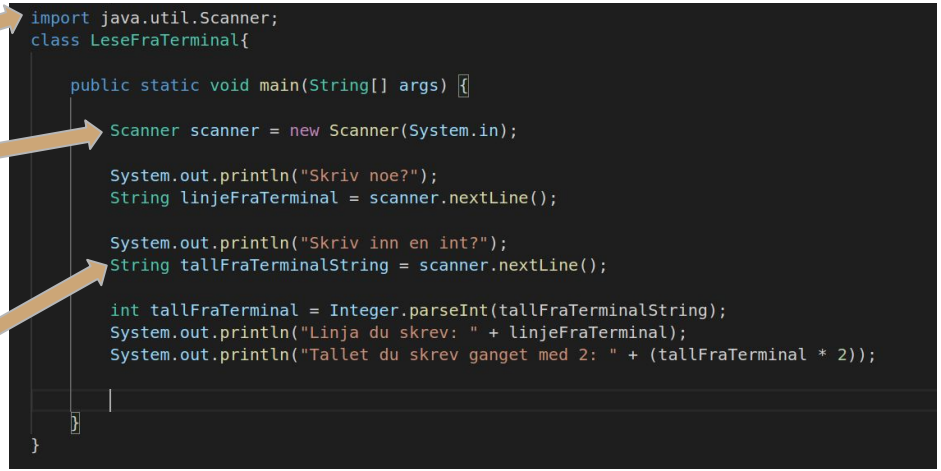
```java
import java.util.ArrayList;

class ArrayListEksempel{

    public static void main(String[] args) {

        ArrayList<String> liste = new ArrayList<>();
        liste.add("Hei ");
        liste.add("paa ");
        liste.add("oss");
        liste.add("!");

        System.out.println(liste.get(0));
        liste.set(2, "deg");
        liste.remove(liste.size() - 1);

        System.out.println(liste);

        if(liste.isEmpty()){
            System.out.println("Listen er tom");
        }else{
            System.out.println("Listen er ikke tom");
        }
    }
}
```

Enkel skriving og lesing i java (dokument): https://www.uio.no/studier/emner/matnat/ifi/IN1010/v21/notater/lesing-og-skriving-i-java.pdf

Ligger under emneressurser på forsiden på semestersiden

I/O - Lese fra terminal

- Bruker **Scanner**:
  - Dokummentasjon: https://docs.oracle.com/javase/8/docs/api/java/util/Scanner.html
  - NB: Må importeres med: import java.util.Scanner
- Sender inn **System.in** i konstruktøren
  - Det forteller Scanner at den skal “lytte til terminalen”
- Nyttige metoder:
  - nextLine(): henter den neste Stringen (til du trykker enter i terminalen)



```
import java.util.Scanner;
class LeseFraTerminal{

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.println("Skriv noe?");
        String linjeFraTerminal = scanner.nextLine();

        System.out.println("Skriv inn en int?");
        String tallFraTerminalString = scanner.nextLine();

        int tallFraTerminal = Integer.parseInt(tallFraTerminalString);
        System.out.println("Linja du skrev: " + linjeFraTerminal);
        System.out.println("Tallet du skrev ganget med 2: " + (tallFraTerminal * 2));


    }
}
```

I/O - Lese fra fil

tekstFil.txt

```
1 en
2 to
3 tre
```

- Bruker **Scanner**:
  - Dokummentasjon: https://docs.oracle.com/javase/8/docs/api/java/util/Scanner.html
  - NB: Må importeres med: import java.util.Scanner
- Sender inn **en fil** i konstruktøren
  - Det forteller Scanner at den skal lese en fil
- Nyttige metoder:
  - hasNextLine(): sjekker om det finnes en ny linje, returnerer en boolean
    - hasNextInt: sjekker om det neste er en int, returnerer en boolean
    - hasNextDouble: sjekker om det neste er en Doble, returnerer en boolean
    - osv....
  - nextLine(): henter den neste Stringen fram til en ny linje
    - nextInt(): henter den neste int-en som er tilgjengelig
    - nextDouble(): henter den neste double-en som er tilgjengelig
    - next(): henter neste ord
    - osv...

```java
import java.util.Scanner;
import java.io.File;
import java.util.HashMap;
import java.io.FileNotFoundException;

class LeseFraFil{

    public static void main(String[] args) {

        File fil = new File("tekstFil.txt");

        Scanner scanner = null;
        try {
            scanner = new Scanner(fil);
        } catch (FileNotFoundException e) {
            System.out.println("Fant ikke filen");
            System.exit(1);
        }

        HashMap<Integer, String> intTilString = new HashMap<>();

        while(scanner.hasNextLine()){
            int tall = scanner.nextInt();
            String tekst = scanner.next();

            intTilString.put(new Integer(tall), tekst);
        }

        System.out.println(intTilString);
    }
}
```

I/O - Output

- Skrive til fil
  - PrintWriter
    - Les mer om den her: https://docs.oracle.com/javase/8/docs/api/java/io/PrintWriter.html
- Skrive til terminal:
  - System.out.println()
  - System.out.print()
  - osv....

Exceptions

Metoder du bruker kan kaste feilmeldinger, hvis noe vi ikke ønsker skal skje har skjedd. (Eks. at den ikke finner en fil). Dette må vi håndtere for at koden ikke skal kræsje. Består av 3 deler:

- try
  - Den delen av koden du vil prøve å kjøre
- catch
  - her spesifiserer du en feilmelding
  - hvis den feilmelding blir oppdaget i try delen vil denne kode blokken kjøre
- finally
  - Det du ønsker at skal skje til slutt uansett
  - Vi kommer mer tilbake til denne senere

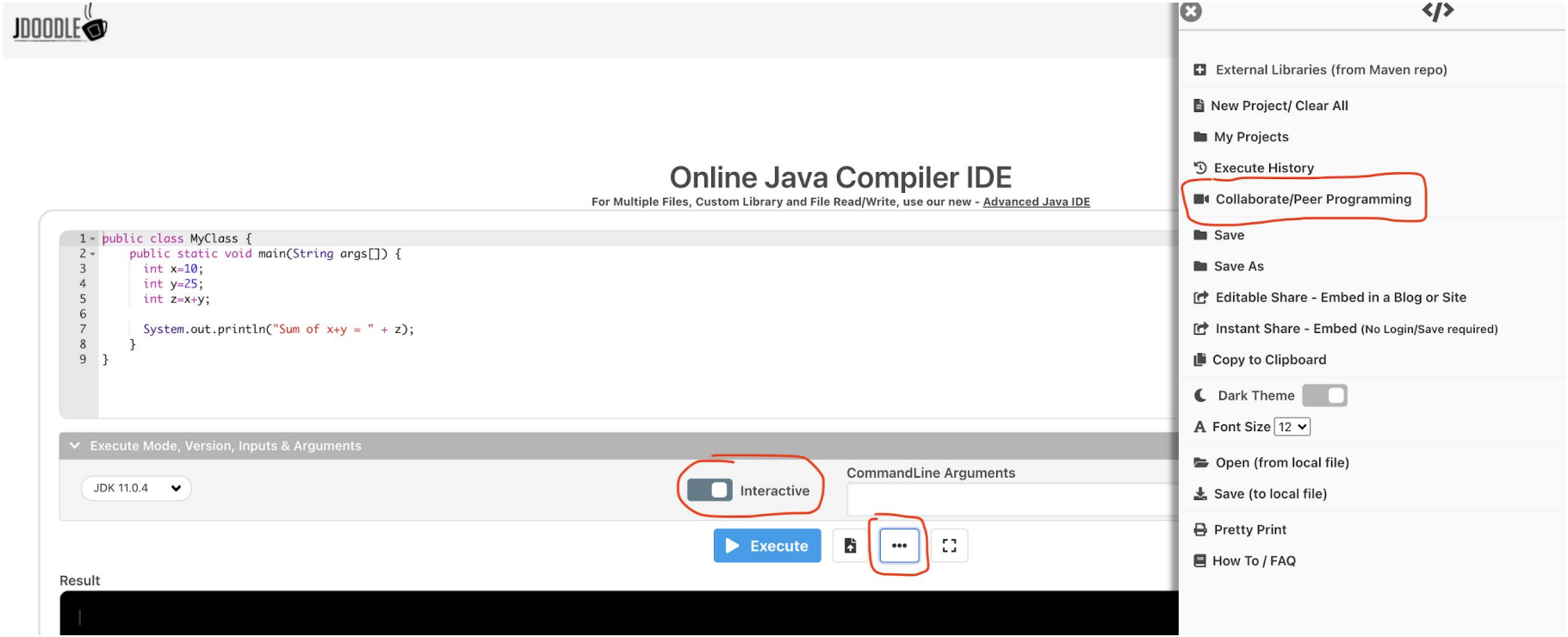
```java
import java.util.Scanner;
import java.io.File;
import java.util.HashMap;
import java.io.FileNotFoundException;

class LeseFraFil{

    public static void main(String[] args) {

        File fil = new File("tekstFil.txt");

        Scanner scanner = null;
        try {
            scanner = new Scanner(fil);
        } catch (FileNotFoundException e) {
            System.out.println("Fant ikke filen");
            System.exit(1);
        }

        HashMap<Integer, String> intTilString = new HashMap<>();

        while(scanner.hasNextLine()){
            int tall = scanner.nextInt();
            String tekst = scanner.next();

            intTilString.put(new Integer(tall), tekst);
        }

        System.out.println(intTilString);
    }
}
```

Jobbe med oppgaver

Breakoutrooms

1. Slå på kamera og ha en presentasjonsrunde
2. Jobb sammen med oppgavene dere finner på mattermost
   a. Enten ved at én deler skjerm
   b. Eller ved å bruke www.gus.jp (velg Java øverst til venstre)
   c. Eller jdoodle.com
   d. OBS: både gus og jdoodle er gratistjeneseter som UiO ikke har avtale med, sannsynligvis vil de samle data om dere. Dere kan fint løse oppgavene uten å bruke disse tjenestene!
3. Bruk “ask for help”-knappen for å få hjelp 😊
4. Vi møtes her igjen for å gå gjennom oppgavene til slutt (dere bestemmer hvilke)

Jdoodle


JDOODLE
Online Java Compiler IDE
For Multiple Files, Custom Library and File Read/Write, use our new - Advanced Java IDE
public class MyClass {
public static void main(String args[]) {
int x=10;
int y=25;
int z=x+y;
System.out.println("Sum of x+y = " + z);
}
}
Execute Mode, Version, Inputs & Arguments
JDK 11.0.4
Interactive
CommandLine Arguments
Execute
Result
External Libraries (from Maven repo)
New Project/ Clear All
My Projects
Execute History
Collaborate/Peer Programming
Save
Save As
Editable Share - Embed in a Blog or Site
Instant Share - Embed (No Login/Save required)
Copy to Clipboard
Dark Theme
Font Size 12
Open (from local file)
Save (to local file)
Pretty Print
How To / FAQ

Ris, ros, forslag ?

https://nettskjema.no/a/181772