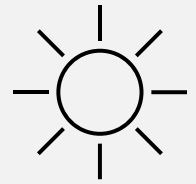
GRUPPE 8 IN1010

Sivert Fjeldstad Madsen

I DAG

- Opplegg for gruppetimer
- IN1010
- Studieteknikker
- Java
- Oppgaver



OPPLEGG FOR GRUPPETIMER

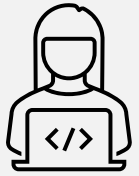
- Sivert Fjeldstad Madsen – sivertfm@uio.no
- Hver mandag 10:15 på C
- Gjennomgang av ukas pensum
  - Nye perspektiver?
- Oppgaver
  - Løses alene og i grupper
- Ikke et sted for å få hjelp til obliger
  - Gå på lab!

IN1010

- Semestersiden
- Java - objektorientering for alvor!
- Krever en del arbeid
- Mye tegning
  - Også på eksamen
- Viktig å ikke henge bakpå på starten

STUDIETEKNIKK

- Skriv kode **hver** dag!
- Prøv å oppnå forståelse
- Vær oppdatert
- Bruk de tilgjengelige ressursene
  - Gruppetimer, labtimer, ukesoppgaver, Trix
- Oppsøk hjelp tidlig

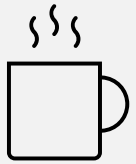


HJELPEHIERARKIET

- Hvem bør du be om hjelp først?
  - Deg selv
  - Pensum
  - En venn
  - Discourse
  - Gruppelærer/Studieadmin
  - Høyere makter

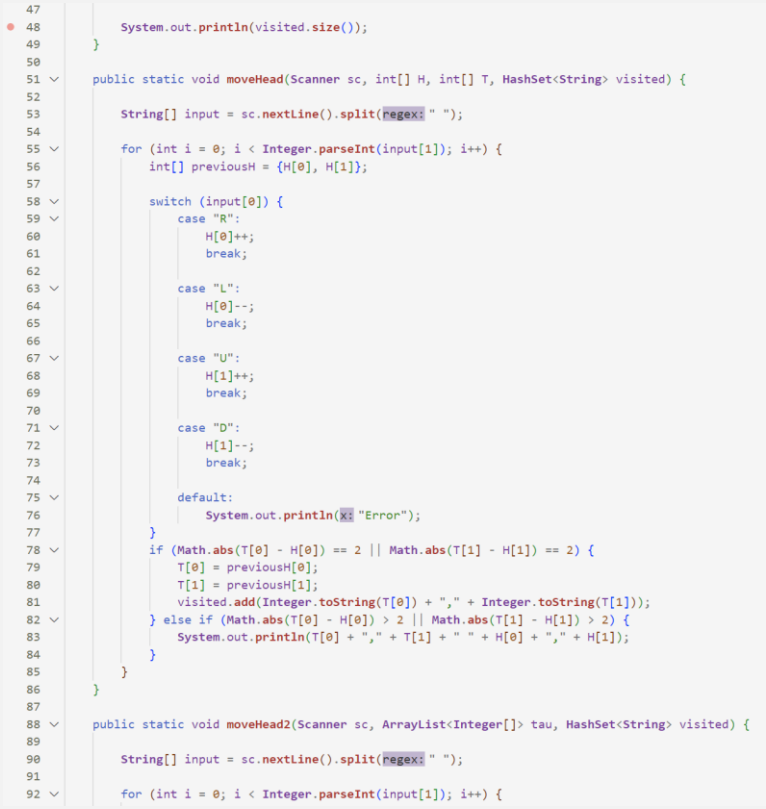
JAVA

- Objektorientering på steroider
- Mer «strengt» enn Python
  - Alt må skrives i klasser
  - Mer syntaks å forholde seg til
  - Et «typet» språk
  - Må kompileres*



GRUNNLEGGENDE SYNTAKS

- class Filnavn {}
- Variabler må deklareres med type:
  - int tall;
  - String navn = “Dag”;
  - boolean javaErMoro = true;
- Semikolon på slutten av hver linje



```
47
48          System.out.println(visited.size());
49      }
50
51      public static void moveHead(Scanner sc, int[] H, int[] T, HashSet<String> visited) {
52
53          String[] input = sc.nextLine().split(regex: " ");
54
55          for (int i = 0; i < Integer.parseInt(input[1]); i++) {
56              int[] previousH = {H[0], H[1]};
57
58              switch (input[0]) {
59                  case "R":
60                      H[0]++;
61                      break;
62
63                  case "L":
64                      H[0]--;
65                      break;
66
67                  case "U":
68                      H[1]++;
69                      break;
70
71                  case "D":
72                      H[1]--;
73                      break;
74
75                  default:
76                      System.out.println(x: "Error");
77              }
78              if (Math.abs(T[0] - H[0]) == 2 || Math.abs(T[1] - H[1]) == 2) {
79                  T[0] = previousH[0];
80                  T[1] = previousH[1];
81                  visited.add(Integer.toString(T[0]) + "," + Integer.toString(T[1]));
82              } else if (Math.abs(T[0] - H[0]) > 2 || Math.abs(T[1] - H[1]) > 2) {
83                  System.out.println(T[0] + "," + T[1] + " " + H[0] + "," + H[1]);
84              }
85          }
86      }
87
88      public static void moveHead2(Scanner sc, ArrayList<Integer[]> tau, HashSet<String> visited) {
89
90          String[] input = sc.nextLine().split(regex: " ");
91
92          for (int i = 0; i < Integer.parseInt(input[1]); i++) {
```

PRIMITIVE TYPER

- I Java er «alt» objekter, bortsett fra de primitive typene
- Vi har **8** primitive typer i Java
  - Heltallstyper:
    - byte
    - short
    - int
    - long
  - Flyttallstyper:
    - float
    - double
  - Andre:
    - char
    - boolean

JAVA-BESVERGELSEN

public static void main(String[] args)

- …også kjent som main-metoden
- Alle Java-prosjekter må ha minst én!
- Er det som blir kjørt når du skriver "java Filnavn" i terminalen

PUBLIC STATIC VOID **MAIN(STRING[] ARGS)**

- «**main**» er **navnet** på metoden
- «String[] args» er parameteren til metoden
  - "String[]" betyr et array med strenger
  - "args" er navnet på variabelen
  - Dette er hvordan du kan få tak i argumenter til programmet ditt!
  - Java krever at du har med dette selv om du ikke skal bruke det til noe

PUBLIC STATIC **VOID** MAIN(STRING[] ARGS)

- «**void**» er **returtypen** til metoden
- Betyr at metoden ikke har noen returverdi
- main-metoden skal aldri returnere noe
- Når du lager egne metoder og funksjoner må du alltid oppgi hvilken type objekt de returnerer

PUBLIC STATIC VOID MAIN(STRING[] ARGS)

- «**public**» er **tilgangsnivået** til metoden
- Vi bruker hovedsakelig to nivåer i IN1010
  - public, som betyr at «alle» har tilgang
  - private, som betyr at kun kode i samme klasse har tilgang
    - Det samme som understrek foran variabel- eller metodenavn i Python
    - Men Java håndhever dette strengt!
- Finnes flere nivåer som ikke er relevante for IN1010

PUBLIC **STATIC** VOID MAIN(STRING[] ARGS)

- «**static**» betyr at du kan bruke metoden (eller variabelen) uten å opprette en instans av klassen
- Vanlige ikke-statiske metoder er avhengige av å bli kalt «på» et objekt!
- Statiske metoder kalles også **klassemetoder**, og tilsvarende kalles statiske variabler **klassevariabler**
  - Disse står da henholdsvis til **instansmetoder** og **instansvariabler**
- Med statiske variabler og metoder kan vi f.eks telle hvor mange instanser av et objekt som har blitt opprettet

STATIC - EKSEMPEL

- Si at vi har en klasse «Person» med metoden «public void siHei()»
- For å kunne bruke metoden, må vi først ha et Person-objekt, altså en **instans** av klassen Person.
  - Person testPerson = new Person();
  - testPerson.siHei();
- Hadde metoden derimot vært «public static void siHei()», hadde vi ikke trengt en instans av klassen.
  - Person.siHei();

Å LÆRE JAVA

- Skriv masse kode
- Prøv deg frem!
- Du lærer utrolig mye av å gjøre feil
- Lær deg å lese og forstå feilmeldinger
- Diskuter problemer og løsninger med andre
  - …men ikke del obligkode!
  - Plagiat kan føre til utestengelse og tap av eksamensrett
- Vær nysgjerrig og utforsk muligheter!

NYTTIGE RESSURSER

- Kodestil i Java
- Objekter og klasser (med tegneeksempler!)
- W3Schools
- Javadokumentasjonen