IN1010 - OBJEKTORIENTERT PROGRAMMERING

# VELKOMMEN TIL ÅTTENDE REPETISJONSTIME

Timen starter 12:15

# REPETISJON

- Comparable
- Lese fra fil
- Skrive til fil
- Unntak og unntakshåndtering
- Iterable og Iterator

# EKSEMPLER OG OPPGAVER

- Comparable<T>-interfacet tilbyr metoden compareTo(T o)
- En klasse kan implementere interfacet for å gjøre objekter av klassen sammenlignbare
- compareTo-metoden returnerer en negativ verdi, 0 eller en positiv verdi

- a.compareTo(b)
  - a < b  gir negativ verdi
  - a == b gir 0
  - a > b  gir positiv verdi

# REPETISJON

- Comparable
- **Lese fra fil**
- Skrive til fil
- Unntak og unntakshåndtering
- Iterable og Iterator

# EKSEMPLER OG OPPGAVER

`nextLine();` `nextInt();` `next();` `nextDouble();`

```
String informasjon = "Henrik 32 1,85 Liker turer i fjellet\n";
Scanner les = new Scanner(informasjon);

String navn = les.

int alder = les.

double hoyde = les.

les.skip(" ");

String bio = les.

les.close();

System.out.print("Profil\n");
System.out.print(navn + ", " + alder + " år gammel, ");
System.out.println(hoyde + " meter høy");
System.out.println(bio);
```

# REPETISJON

- Comparable
- **Lese fra fil**
- Skrive til fil
- Unntak og unntakshåndtering
- Iterable og Iterator

# EKSEMPLER OG OPPGAVER

`nextLine();` `nextInt();` `next();` `nextDouble();`

```
String informasjon = "Henrik 32 1,85 Liker turer i fjellet\n";
Scanner les = new Scanner(informasjon);

String navn = les.next();

int alder = les.nextInt();

double hoyde = les.nextDouble();

les.skip(" ");

String bio = les.nextLine();

les.close();

System.out.print("Profil\n");
System.out.print(navn + ", " + alder + " år gammel, ");
System.out.println(hoyde + " meter høy");
System.out.println(bio);
```

# REPETISJON

- Comparable
- **Lese fra fil**
- Skrive til fil
- Unntak og unntakshåndtering
- Iterable og Iterator

# EKSEMPLER OG OPPGAVER

- **Først kan en opprette et File-objekt**
- **Så kan en sende fila til et Scanner-objekt**
- **Her er det viktig å sjekke om det finnes noe å skanne før man gjør det!**

```
File fil = new File("tekstfil.txt");
Scanner les = new Scanner(fil);

les.hasNext();
les.hasNextLine();

les.hasNextInt();
les.hasNextDouble();
```

# REPETISJON

- Comparable
- Lese fra fil
- **Skrive til fil**
- Unntak og unntakshåndtering
- Iterable og Iterator

# EKSEMPLER OG OPPGAVER

- For å skrive ut til fil kan en benytte et PrintWriter-objekt
- Man sender med filnavnet det skal skrives til, og så kan en bruke print-, println- og printf-metodene for å skrive til filen
- Husk å lukke PrintWriter-objektet med close-metoden



```
PrintWriter skriv = new PrintWriter("profil.txt");

skriv.print("Profil\n");
skriv.print(navn + ", " + alder + " år gammel, ");
skriv.println(hoyde + " meter høy");
skriv.println(bio);

skriv.close();
```

# REPETISJON

- Comparable
- Lese fra fil
- Skrive til fil
- **Unntak og unntakshåndtering**
- Iterable og Iterator

# EKSEMPLER OG OPPGAVER

- **Handler om å oppdage feil og håndtere feil**
- **Feil oppdaget: throw new unntak**
- **Velg passende type feilmelding**
- **Programmet stopper opp der hvor unntaket kastes**

- **Håndtere feil: bruk catch**
- **Catch spesifikt først, og så mer generelt**
- **Ingen handler gir feilmelding og programmet avsluttes**
- **Kan kaste videre med throws**

# REPETISJON

- Comparable
- Lese fra fil
- Skrive til fil
- Unntak og unntakshåndtering
- **Iterable og Iterator**

# EKSEMPLER OG OPPGAVER

- **For å kunne ta i bruk en forenklet for-løkke for å iterere over en lenkeliste, kan en benytte seg av intefacene Iterable<T> og Iterator<T>**
- **Klassen det skal itereres over implementerer Iterable<T> og må dermed override'e metoden iterator( )**
- **Så kan man ha en indre iterator-klasse som er spesifikk for klassen den er inni**
- **Den indre klassen implementerer Iterator<T> og må dermed override'e metodene hasNext( ) og next( )**

# VI SEES

HADE 👋