IN1010 gruppetime

25.03.2026

Plan for dagen

Gjennomgang av forrige eksamen

FIB

Intro til tråder

Gjennomgang av eksamen!

FIB



peoply.app

Peoply

Integrasjoner FAQ **Logg in**

Kalender

Åpne kalender

Hva skjer fremover?

Se alle



I DAG 12:00

Stikkball turnering med IFI-RASTLØS

6

IFI Rastløs

gymsalen på Athletica Blindern

Legg i kalender



I DAG 16:15

Git Good – Andreåringer

14/80

FUI

Seminarrom C

Legg i kalender



I DAG 16:15

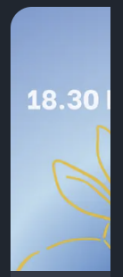
Genfors Maki

6

MA:KI

Smalltalk

Legg i kalender



I DAG

QUIZ

Q

E

Foreninger på IFI

Se alle

Hovedsamarbeidspartner

Bekk

Bekk er et konsulentselskap med litt over 600 ansatte. En gjeng teknologer, designere og rådgivere som liker å løse problemer. Hvordan vi løser dem varierer, men fellesnevneren er alltid den samme: folk med ulik bakgrunn og kompetanse tenker sammen, utfyller hverandre, utfordrer hverandre og de vi jobber med. Vi tror et inkluderende miljø som bringer frem det beste i folk, faglig og sosialt, er et bra sted å jobbe. Vi mener at vi presterer bedre i team enn alene, og at faglig utvikling er noe vi gjør sammen, ikke på egen hånd. Du finner oss ytterst på Vippetangen i Oslo!

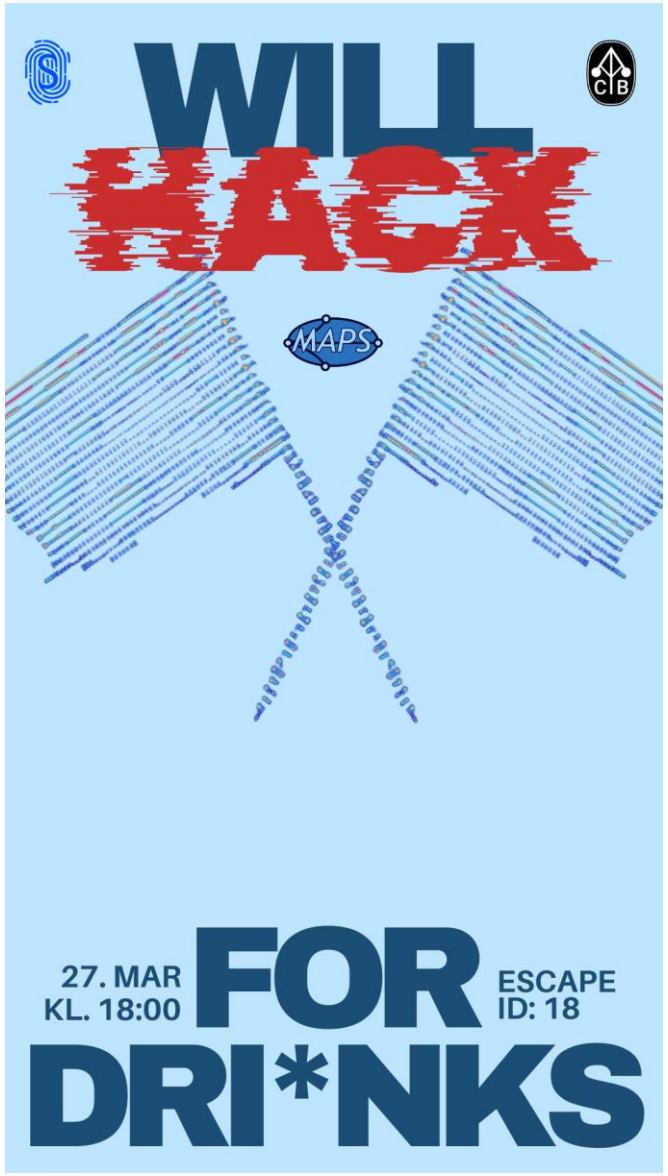
Se våre stillingsannonser

Bedriftspresentasjon med PwC

Velkommen til kurs med PwC!

40 torsdag 26. mars

Se alle arrangementer


WILL
HACK
MAPS
27. MAR
KL. 18:00
FOR
ESCAPE
ID: 18
DRI*NKS

Tråder

Call-stacken

```java
class HovedProgram {
    Run | Debug
    public static void main(String[] args) {

        Hund hund1 = new Hund();
        hund1.gåTur();
    }
}

class Hund {

    void gåTur() {
        System.out.println(x: "Jeg går.");
    }
}
```

Call-stack:

```java
class HovedProgram {
    Run | Debug
    public static void main(String[] args) {

        Hund hund1 = new Hund();
        hund1.gåTur();
    }
}

class Hund {

    void gåTur() {
        System.out.println(x: "Jeg går.");
    }
}
```


Call-stack:



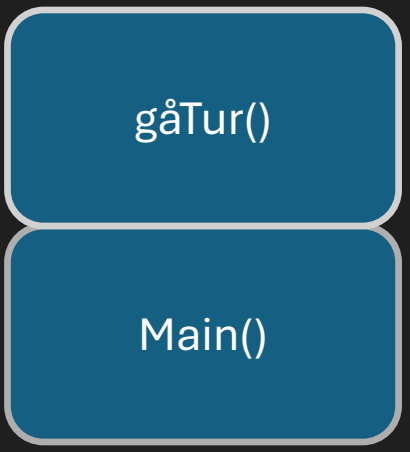
```java
class HovedProgram {
    Run | Debug
    public static void main(String[] args) {

        Hund hund1 = new Hund();
        hund1.gåTur();
    }
}

class Hund {

    void gåTur() {
        System.out.println(x: "Jeg går.");
    }
}
```

Call-stack:



```java
class HovedProgram {
    Run | Debug
    public static void main(String[] args) {

        Hund hund1 = new Hund();
        hund1.gåTur();
    }
}

class Hund {

    void gåTur() {
        System.out.println(x: "Jeg går.");
    }
}
```



Call-stack:

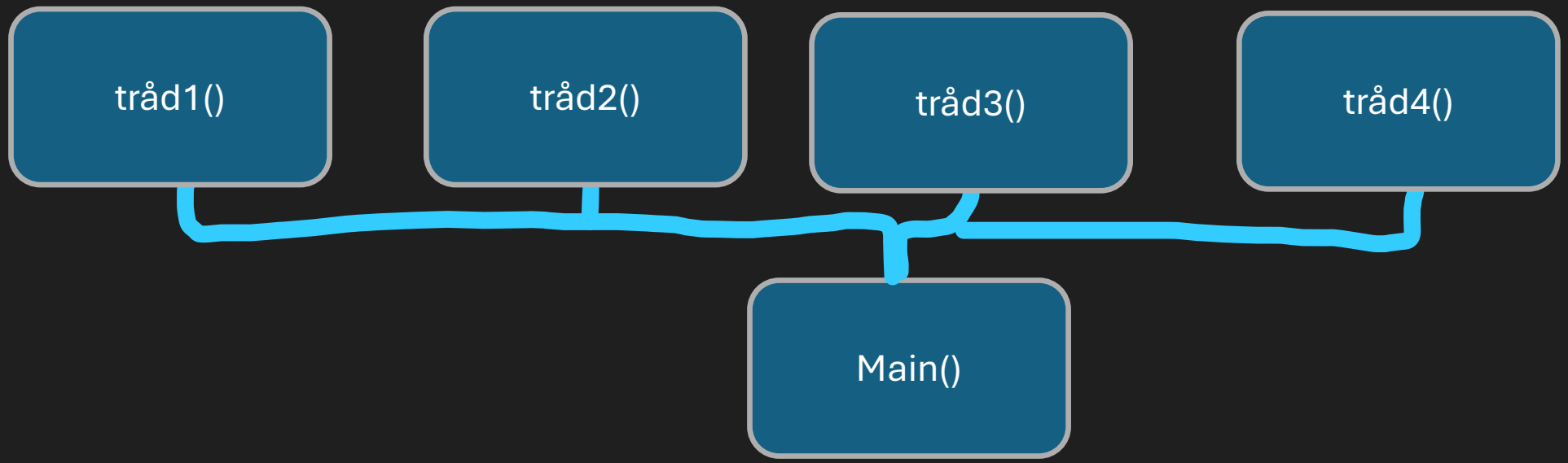


Okei bro, skjønner hva en call-stack er...

...men hva er tråder??

Hvordan skrive java-kode til dette?

Call-stack:



Tråd-klasser

*Implementerer *runnable*

Representerer oppgaver – ikke fysiske objekter

Har en konstruktør og en run-metode

Tråd-klasser

```
class GåTurTråd implements Runnable {
    private Hund hund;

    public GåTurTråd(Hund hund) {
        this.hund = hund;
    }

    @Override
    public void run() {
        System.out.println(hund + "Denne tråden går tur!");

    }
}
```

Hvordan bygge call-stack med tråder?

```java
public class Main {
    Run | Debug
    public static void main(String[] args) {
        Thread hundegåer1 = new Thread(new GåTurTråd(new Hund()));
        Thread hundegåer2 = new Thread(new GåTurTråd(new Hund()));
        Thread hundegåer3 = new Thread(new GåTurTråd(new Hund()));

        hundegåer1.start();
        hundegåer2.start();
        hundegåer3.start();
    }
}

class GåTurTråd implements Runnable {
    private Hund hund;

    public GåTurTråd(Hund hund) {
        this.hund = hund;
    }

    @Override
    public void run() {
        System.out.println(hund + "Denne tråden går tur!");

    }
}
```


Call-stack:



```java
public class Main {
    Run | Debug
    public static void main(String[] args) {
        Thread hundegåer1 = new Thread(new GåTurTråd(new Hund()));
        Thread hundegåer2 = new Thread(new GåTurTråd(new Hund()));
        Thread hundegåer3 = new Thread(new GåTurTråd(new Hund()));

        hundegåer1.start();
        hundegåer2.start();
        hundegåer3.start();
    }
}

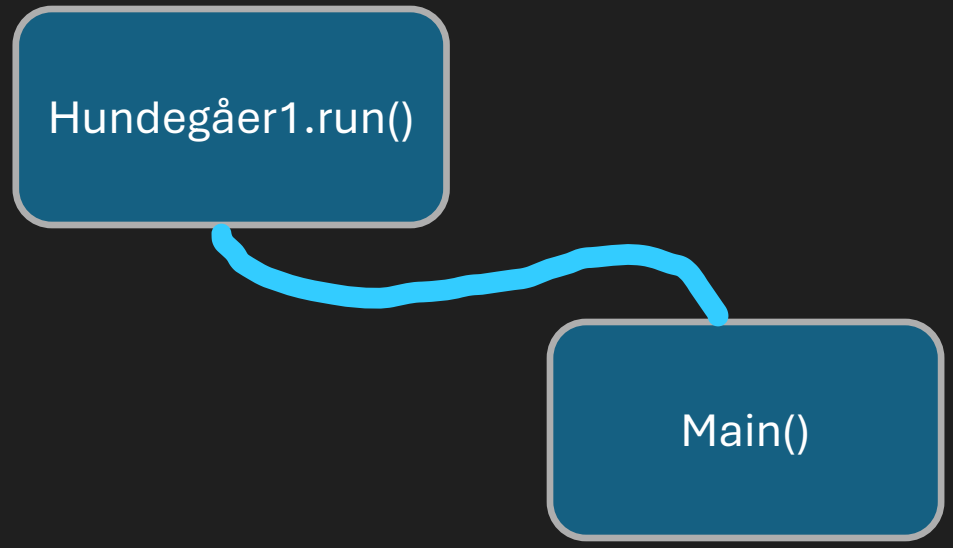
class GåTurTråd implements Runnable {
    private Hund hund;

    public GåTurTråd(Hund hund) {
        this.hund = hund;
    }

    @Override
    public void run() {
        System.out.println(hund + "Denne tråden går tur!");

    }
}
```

Call-stack:



```java
public class Main {
    Run | Debug
    public static void main(String[] args) {
        Thread hundegåer1 = new Thread(new GåTurTråd(new Hund()));
        Thread hundegåer2 = new Thread(new GåTurTråd(new Hund()));
        Thread hundegåer3 = new Thread(new GåTurTråd(new Hund()));

        hundegåer1.start();
        hundegåer2.start();
        hundegåer3.start();
    }
}

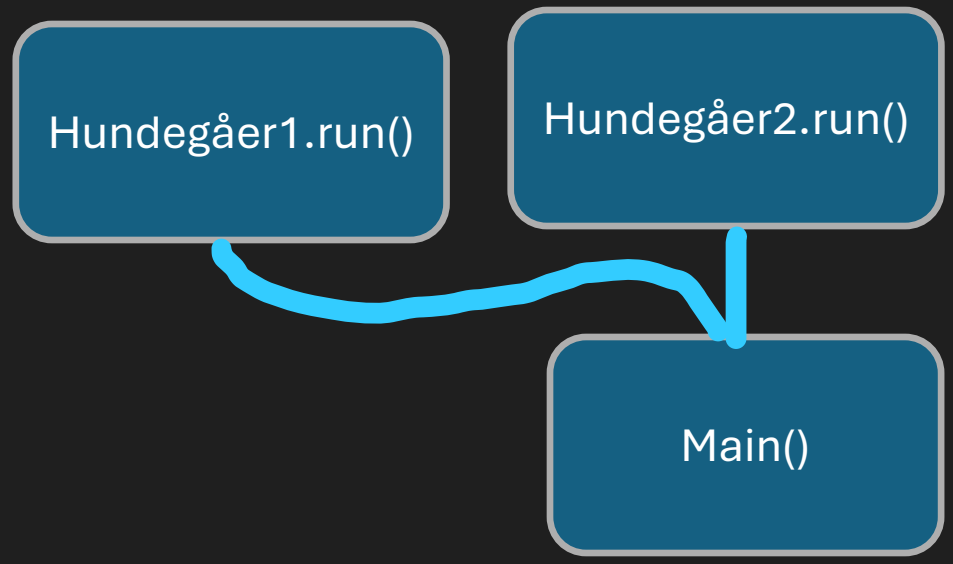
class GåTurTråd implements Runnable {
    private Hund hund;

    public GåTurTråd(Hund hund) {
        this.hund = hund;
    }

    @Override
    public void run() {
        System.out.println(hund + "Denne tråden går tur!");

    }
}
```

Call-stack:



```java
public class Main {
    Run | Debug
    public static void main(String[] args) {
        Thread hundegåer1 = new Thread(new GåTurTråd(new Hund()));
        Thread hundegåer2 = new Thread(new GåTurTråd(new Hund()));
        Thread hundegåer3 = new Thread(new GåTurTråd(new Hund()));

        hundegåer1.start();
        hundegåer2.start();
        hundegåer3.start();
    }
}

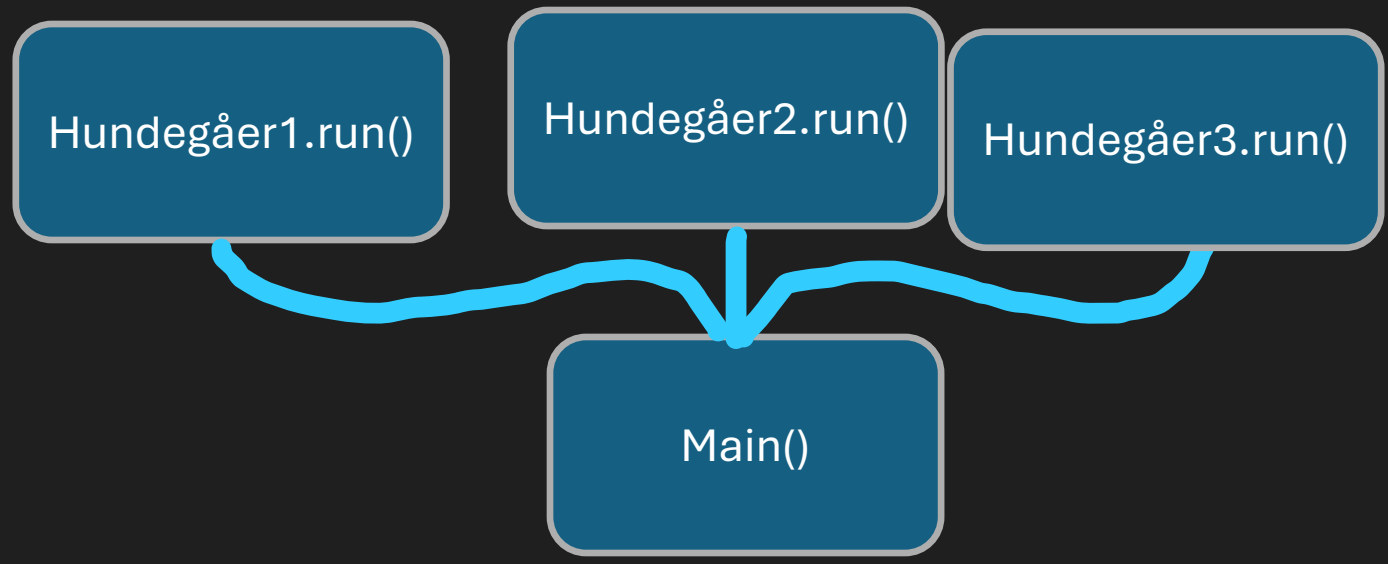
class GåTurTråd implements Runnable {
    private Hund hund;

    public GåTurTråd(Hund hund) {
        this.hund = hund;
    }

    @Override
    public void run() {
        System.out.println(hund + "Denne tråden går tur!");

    }
}
```

Call-stack:



Andre nyttige metoder til tråder:

thread.join()	**Brukes for å vente til alle tråder er ferdige*
thread.sleep(X)	**Brukes for å «sove» i X antall millisekunder*
thread.interrupt(X)	**Forteller tråden at den skal stoppe*

Mer om tråder neste gang:

Monitorer

Skrive koden trådsikkert

Takk for meg!!