Oppgave 1 - Klasser og objekter

```
class Kaningaard {
    private Kanin [] kaniner = new Kanin[100];

    public boolean full() {
        for (int i = 0; i < kaniner.length; i++) {
            if (kaniner[i] == null) {
                return false;
            }
        }
        return true;
    }

    public boolean tom () {
        for (int i = 0; i < kaniner.length; i++) {
            if (kaniner[i] != null) {
                return false;
            }
        }
        return true;
    }

    public Kanin finnEn(String navn) {
        for (int i = 0; i < kaniner.length; i++) {
            if (kaniner[i] != null && kaniner[i].hentNavn().equals(navn)) {
                return kaniner[i];
            }
        }
        return null;
    }

    public void settInn (Kanin kanin) {

        if (finnEn(kanin.hentNavn()) != null) {
            System.out.println("Kaninen finnes alt i kaningaarden!");
        } else if (full()) {
            System.out.println("Det er ikke plass til flere kaniner!");
        } else {

            boolean sattInn = false;
            int teller = 0;

            while (teller < kaniner.length && !sattInn) {
                if (kaniner[teller] == null) {
                    kaniner[teller] = kanin;
                    sattInn = true;
                }
                teller++;
            }
        }
    }

    public void fjern(String navn) {
        boolean funnet = false;
        int teller = 0;

        while (teller < kaniner.length && !funnet) {
            if (kaniner[teller] != null && kaniner[teller].hentNavn().equals(navn)) {
                kaniner[teller] = null;
                funnet = true;
            }
            teller++;
        }
```

```
        if (!funnet) {
            System.out.println("Fant ikke kaninen " + navn + ".");
        }
    }

    //Et par tester ...
    public static void main(String[] args) {
        Kaningaard kg = new Kaningaard();
        kg.settInn(new Kanin("Kaare"));
        kg.settInn(new Kanin("Kaare"));
        kg.fjern("Kaare");
        kg.fjern("Kaare");
    }
}
```

Oppgave 2 - Robotfamilien

```
abstract class Robot {
    private String name;
    public Robot(String name) {
        this.name = name;
    }
    public String getName() {
        return name;
    }
}

class GPT3 extends Robot {
    private static ArrayList<String> sharedProperties = new ArrayList<>();
    private ArrayList<String> uniqueProperties;
    public GPT3(String name) {
        super(name);
        this.uniqueProperties = new ArrayList<>();
    }

    public void addSharedProperty(String property) {
        sharedProperties.add(property);
    }
    public void removeSharedProperty(String property) {
        sharedProperties.remove(property);
    }

    public void addUniqueProperty(String property) {
        uniqueProperties.add(property);
    }
    public void removeUniqueProperty(String property) {
        uniqueProperties.remove(property);
    }
}
```

```java
class GPT35 extends GPT3 {
    public GPT35(String name) {
        super(name);
    }
}

class GPT4 extends GPT3 {
    public GPT4(String name) {
        super(name);
    }
}
```

Oppgave 3 - Arv og HashMaps

```java
import java.util.HashMap;

class Kjøretøy {
    private String id;
    private String merke;
    private String modell;
    private int år;

    public Kjøretøy(String id, String merke, String modell, int år) {
        this.id = id;
        this.merke = merke;
        this.modell = modell;
        this.år = år;
    }

    public String hentId() {
        return id;
    }

    public String hentMerke() {
        return merke;
    }

    public void settMerke(String merke) {
        this.merke = merke;
    }

    public String hentModell() {
        return modell;
    }
```

```
    public void settModell(String modell) {
        this.modell = modell;
    }

    public int hentÅr() {
        return år;
    }
    public void settÅr(int år) {
        this.år = år;
    }
}

class Bil extends Kjøretøy {
    private int seter;

    public Bil(String merke, String modell, int år, int seter) {
        super(merke, modell, år);
        this.seter = seter;
    }

    public int hentSeter() {
        return seter;
    }

    public void settSeter(int seter) {
        this.seter = seter;
    }
}

class Motorsykkel extends Kjøretøy {
    private String motortype;

    public Motorsykkel(String merke, String modell, int år, String motortype) {
        super(merke, modell, år);
        this.motortype = motortype;
    }

    public String hentMotortype() {
        return motortype;
    }

    public void settMotortype(String motortype) {
        this.motortype = motortype;
    }
}


class Hovedprogram {
```

```
    public static void main(String[] args) {
        HashMap<String, Kjøretøy> kjøretøyMap = new HashMap<>();

        Bil bil1 = new Bil("Mercedes", "C-Klasse", 2020, 5);
        kjøretøyMap.put(bil1.hentId(), bil1);

        Bil bil2 = new Bil("BMW", "5-Serie", 2019, 5);
        kjøretøyMap.put(bil2.hentId(), bil2);

        Motorsykkel motorsykkel1 = new Motorsykkel("Honda", "CBR600RR", 2018, "Sport");
        kjøretøyMap.put(motorsykkel1.hentId(), motorsykkel1);

        Motorsykkel motorsykkel2 = new Motorsykkel("Yamaha", "YZF-R6", 2021, "Sport");
        kjøretøyMap.put(motorsykkel2.hentId(), motorsykkel2);

        System.out.println("Kjøretøydetaljer:");

        for (Kjøretøy kjøretøy : kjøretøyMap.values()) {
            System.out.println("ID: " + kjøretøy.hentId());
            System.out.println("Merke: " + kjøretøy.hentMerke());
            System.out.println("Modell: " + kjøretøy.hentModell());
            System.out.println("År: " + kjøretøy.hentÅr());
            if (kjøretøy instanceof Bil) {
                System.out.println("Seter: " + ((Bil) kjøretøy).hentSeter());
            } else if (kjøretøy instanceof Motorsykkel) {
                System.out.println("Motortype: " + ((Motorsykkel)
kjøretøy).hentMotortype());
            }
            System.out.println();
        }
    }
}
```

Oppgave 4 - Filinnlesning, objekter og arv

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

class SlaaSammenOrdA {
    public static void main(String[] args) {
        File file = new File("ord.txt");
        try {
            Scanner scanner = new Scanner(file);
            String sentence = "";
```

```
            while (scanner.hasNextLine()) {
                sentence += scanner.nextLine() + " ";
            }

            scanner.close();
            System.out.println(sentence);
        } catch (FileNotFoundException e) {
            System.out.println("Filen ble ikke funnet.");
        }
    }
}

class SlaaSammenOrdB {
    public static void main(String[] args) {
        Scanner brukerInput = new Scanner(System.in);
        System.out.print("Skriv inn filnavnet: ");
        String filnavn = brukerInput.nextLine();
        brukerInput.close();

        File file = new File(filnavn);

        try {
            Scanner scanner = new Scanner(file);
            String setning = "";

            while (scanner.hasNextLine()) {
                setning += scanner.nextLine() + " ";
            }

            scanner.close();

            System.out.println(setning);
        } catch (FileNotFoundException e) {
            System.out.println("Filen ble ikke funnet.");
        }
    }
}

class SlaaSammenOrdC {
    public static void main(String[] args) {
        Scanner brukerInput = new Scanner(System.in);
        System.out.print("Skriv inn filnavnet: ");
        String fileName = brukerInput.nextLine();
        brukerInput.close();

        File file = new File(fileName);
```

```
        try {
            Scanner scanner = new Scanner(file);
            String setning = "";

            while (scanner.hasNextLine()) {
                setning += scanner.nextLine() + " ";
            }

            scanner.close();

            System.out.println(setning);
        } catch (FileNotFoundException e) {
            System.out.println("Filen ble ikke funnet.");
        }
    }
}

class SlaaSammenOrdD {
    public static void main(String[] args) {
        Scanner brukerInput = new Scanner(System.in);
        System.out.print("Skriv inn filnavnet: ");
        String filnavn = brukerInput.nextLine();
        brukerInput.close();

        File fil = new File(filnavn);

        try {
            Scanner scanner = new Scanner(fil);
            String setning = "";

            while (scanner.hasNextLine()) {
                String ord = scanner.nextLine();
                setning += ord + " ";
                Ord ordobjekt = new Ord(ord);
                ordobjekt.skrivUt();
            }

            scanner.close();

            System.out.println(setning);
        } catch (FileNotFoundException e) {
            System.out.println("Filen ble ikke funnet.");
        }
    }
}
class Ord {
    private String ord;
```

```
    public Ord(String ord) {
        this.ord = ord;
    }

    public void skrivUt() {
        System.out.println(ord);
    }
}
```

Oppgave 5 - Arrayer, ArrayList og for-løkker

```
import java.util.ArrayList;

public class Main {
    public static void main(String[] args) {
        // Oppgave a
        int[] array = new int[10];
        for (int i = 0; i < array.length; i++) {
            array[i] = i;
            System.out.println(array[i]);
        }

        // Oppgave b
        ArrayList<Integer> list = new ArrayList<>();
        for (int i = 0; i < array.length; i++) {
            list.add(i);
        }

        for (int i = list.size() - 1; i >= 0; i--) {
            System.out.println(list.get(i));
        }

        // Oppgave c
        int sum = 0;
        for (int number : array) {
            System.out.println(number);
            sum += number;
            System.out.println("Sum: " + sum);
        }

        // Oppgave d
        String result = "";
        for (int number : array) {
            result += number;
            System.out.println(result);
```

```
        }
    }
}
```