

1)**c)**

```
public class Rhino extends Animal{
    public Rhino(int nbrOfLegs, String name) {
        super(nbrOfLegs, name);
    }
}
```

d)

```
public class Test {
    public static void main(String[] args){
        Animal zebra1 = new Zebra(4, "Sven");
        //Eller: Zebra zebra1 = new Zebra(4, "Sven");
        System.out.println(zebra1.getName());
        System.out.println(zebra1.getNbrOfLegs());

        Giraffe giraffel = new Giraffe(4, "Yngve", 1.8);
        System.out.println(giraffel.getName());
        System.out.println(giraffel.getNbrOfLegs());
        System.out.println(giraffel.getNeckLength());

        Animal rhinol = new Rhino(4, "Lars-Åke");
        //Eller: Rhino rhinol = new Rhino(4, "Lars-Åke");
        System.out.println(rhinol.getName());
        System.out.println(rhinol.getNbrOfLegs());
    }
}
```

2)**c)**

```
public class Pitbull implements IDog{
    @Override
    public void doTrick() {
        System.out.println("Sit");
    }
}
```

e)

//Med array

```
public class TestWithArray {
    public static void main(String[] arg){
        //create dogs
        IDog poodle = new Poodle();
    }
}
```

```
IDog goldenRetriever = new GoldenRetriever();
IDog doge = new Doge();

//create array
IDog[] dogArray = {poodle, goldenRetriever, doge};

//iterate through all dogs and do their tricks
for(int i = 0; i < dogArray.length; i++){
    dogArray[i].doTrick();
}

}

//Med lista
public class TestWithList {
    public static void main(String[] arg){
        //create dogs
        IDog poodle = new Poodle();
        IDog goldenRetriever = new GoldenRetriever();
        IDog doge = new Doge();

        //create list
        List<IDog> listOfDogs = new ArrayList<IDog>();

        //add dogs to the list
        listOfDogs.add(poodle);
        listOfDogs.add(goldenRetriever);
        listOfDogs.add(doge);

        //iterate through all dogs and do their tricks
        for(int i = 0; i < listOfDogs.size(); i++){
            listOfDogs.get(i).doTrick();
        }
    }
}
```

3)

```
import java.util.Scanner;

public class Model {
    private static int player1NbrOfWins;
    private static int player2NbrOfWins;
    private static int player1ToolChoice;
    private static int player2ToolChoice;
    private static int playOrNoPlayIdentifier;

    private static void play() {
        System.out.println("Player 1's turn");
        player1ToolChoice = getPlayerInput();
        System.out.println("\n\n\n\n\n\n\n\n\n\n\n\n\n");
        System.out.println("Player 2's turn");
        player2ToolChoice = getPlayerInput();
        System.out.println("\n\n\n\n\n\n\n\n\n\n\n\n\n");
        decideWinner();
    }

    private static void decideWinner() {
        if(player1ToolChoice==player2ToolChoice){
            System.out.println("It's a draw!\n");
        }
        else
        if((((player2ToolChoice-player1ToolChoice)%3)+3)%3)==1){
            System.out.println("Player 1 wins!\n");
            player1NbrOfWins++;
        }
        else{
            System.out.println("Player 2 wins!\n");
            player2NbrOfWins++;
        }
        System.out.println("'" + player1NbrOfWins + "-" +
player2NbrOfWins);
    }

    private static int getPlayerInput() {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Choose tool:\n1=Sten\n2=Sax\n3=Pase");
        return scanner.nextLine().charAt(0);
    }
}
```

```

    public static void main(String[] args){
        while(true){
            Scanner scanner = new Scanner(System.in);
            System.out.println("Press 1 to play.\nPress 0 to
quit.");

            playOrNoPlayIdentifier = scanner.nextInt();
            if(playOrNoPlayIdentifier==0){
                break;
            }
            play();
        }
    }
}

```