// overloading

// returntype string int

// aanroep vervangen door dat wat hij return

// returntype object

// parameter object

// argument object

// argument string

// parameter int

class Demo {

public static void main(String[] args){

new Konijn().starten();

Fokker bart = new Fokker();

bart.konijnKopen().eten("Radijsjes");

Konijn piet = new Konijn();

char a = bart.konijnKopen().zijnNaamGeven().charAt(2);

piet.eten("appel");

System.out.println(a);

}

}

class Fokker{

Konijn konijnKopen(){

return new Konijn();

}

}

class Konijn{

void starten(){

System.out.println("Welkom bij het Konijn");

eten("Sla"); // argument

eten("Mais");

String deNaam = zijnNaamGeven();

System.out.println(deNaam);

System.out.println( geefAcht() + geefAcht());

}

int geefAcht(){

return 8;

}

void eten(String soortVoedsel){ // parameter

System.out.println("Het konijn eet een " + soortVoedsel);

}

void eten(int a){

}

String zijnNaamGeven(){

return "Frits";

}

}

=================

class Demo {

public static void main(String[] args){

new Konijn().starten();

Fokker bart = new Fokker();

bart.konijnKopen().eten("Radijsjes");

}

}

class Fokker{

Konijn konijnKopen(){

return new Konijn();

}

}

class Konijn{

void starten(){

System.out.println("Welkom bij het Konijn");

eten("Sla"); // argument

eten("Mais");

String deNaam = zijnNaamGeven();

System.out.println(deNaam);

System.out.println( geefAcht() + geefAcht());

}

int geefAcht(){

return 8;

}

void eten(String soortVoedsel){ // parameter

System.out.println("Het konijn eet een " + soortVoedsel);

}

String zijnNaamGeven(){

return "Flappie";

}

}

==============

class BlackJackApp {

public static void main(String[] args){

new BlackJack().start();

}

}

class BlackJack{

void start(){

System.out.println("Welkom bij blackjack");

}

}

=============

import java.util.Scanner;

class Demo {

public static void main(String[] args){

System.out.println("Welkom bij BlackJack");

String[] deck = {"H2","H3","H4","H5","H6","H7", "H8", "H9", "HB"};

for(int i = 0; i < deck.length; i++){

System.out.print( deck[i] +" " );

}

System.out.println();

System.out.println("Wilt u de kaarten schudden j voor ja n voor nee");

Scanner sc = new Scanner(System.in);

System.out.println(sc.next());

}

}

==========

class Demo {

public static void main(String[] args){

int a = 3;

if(a <= 3 || ++a < 10 ){ // Short circuited

System.out.println("Hoi");

}

System.out.println(a);

}

void lopen(Hond d){

d.pootjes = 25;

}

}

class Hond{

static int ogen = 4;

int pootjes = 6;

}

=================

class Demo {

public static void main(String[] args){

Integer i = new Integer(5);

int j = i; // unboxing

short r = 5;

Integer e = j; // kan wel autoboxing

// Integer z = r; // kan niet

}

void lopen(Hond d){

d.pootjes = 25;

}

}

class Hond{

static int ogen = 4;

int pootjes = 6;

}

====================

class Demo {

public static void main(String[] args){

int a = 8;

Hond hond;

Hond hond2 = new Hond();

Hond hond3 = new Hond();

hond2.ogen = 5;

System.out.println(hond3.ogen);

}

}

class Hond{

static int ogen;

int pootjes;

}

=================

BlackJack, ook wel 21-en genoemd.

<https://nl.wikipedia.org/wiki/Blackjack>

De Black Jack wordt in de commandline gemaakt.

Het zal een werkend spel moeten zijn bij oplevering.

Doordat het een commandline spel is mag je natuurlijk creatief zijn in de interactie met de gebruiker. H4 kan een harten 4 zijn. Zolang de onderliggende techniek maar werkt.

Op de wikipedia pagina zie je dat er heel veel regels, uitbreidingen en varianten zijn voor een compleet BlackJack spel.

De kunst is als programmeur om deze op te knippen en met kleine doelstellingen te beginnen.

Dat werk heb ik al een beetje voor je gedaan. Hou daarom als leidraad de volgende doelstellingen aan:

Doelstelling 1

Elke Black Jack begint met het tonen van de geschudde kaarten.

De Black Jack spelen we met 1 spel van 52 kaarten. Alle kaarten zitten dus 1 keer in het overzicht, en als het spel opnieuw gestart wordt is de volgorde in een random nieuwe volgorde.

Doelstelling 2

Een speler kan meedoen met het spel. De speler kan door een 'k' in te voeren om een kaart vragen. Een speler kan altijd met heel het spel stoppen door 'q' in te voeren. De speler mag ook 'p' invoeren om te passen.

Als een speler een kaart vraagt krijgt hij de bovenste kaart van de stapel. Het totaal aantal punten wordt getoont.

De speler mag met 0 kaarten starten. De aas mag als een fixed 11 punten kaart beschouwd worden.

Als de speler past, worden alle kaarten getoont die hij heeft met het totaal aantal punten van de kaarten.

De punten per kaart zijn: De waarde van het getal bij de 2, 3, 4, 5, 6, 7, 8, 9, 10. De Boer, Vrouw, Heer zijn 10 punten. De aas is zoals gezegd 11 punten.

Doelstelling 3

Vanaf hier wordt het spannender. Het is vanaf deze doelstelling verplicht om met kaarten als objecten te werken. Kaart kaart = new Kaart(). De vorige doelstellingen konden prima op andere wijzen volbracht worden, vanaf hier is Object Georienteerd programmeren een vereiste.

De speler krijgt twee kaarten toebedeeld wanneer hij start, zoals bij de werkelijke regels. Ook kan de Aas 1 of 11 punten waard zijn. Dat is afhankelijk van het gegeven of hij anders boven de 21 punten uit komt.

Als een speler niet op tijd past, is de speler af.

Er mag geen invoer zijn die het programma doet vastlopen.

Mocht je tijd over hebben dan kun je je bezig gaan houden met meerdere spelers, een bank of zelfs gokken en inzetten.

=========================

501 ls

502 pwd

503 cd ..

504 cd ..

505 cd \_FILES/

506 cd y

507 cd ycLeiden/

508 ls

509 history

510 clear

511 git init

512 git init

513 git status

514 git add .

515 git status

516 git commit -am "eerste commit"

517 git status

518 git log

519 git status

520 git add .

521 git commit -am "tweede foto"

522 git status

523 git log

524 git remote add origin https://github.com/FelixvL/trialleiden1.git

525 git push

526 git push -u origin master

527 history

=========================

dylanbonga@gmail.com

joeynapy@gmail.com

joshua.hully@gmail.com

justuswolbert@gmail.com

irenedekoster@gmail.com

161108jr@gmail.com

DylanX

https://github.com/DylanBonga

JoeyX

https://github.com/joeynap

JoshuaX

https://github.com/JoshuaDaoud

JustusX

https://github.com/jlwwolbert

TreentjeX

https://github.com/Treentje

jimX

Typ hier uw programma.https://github.com/JamesReginald/