

```

//@author maximilian raspe
import java.util.Scanner;
public class Wochentag {

    public static void main(String[] args){

        String wochentag;
        Scanner scan = new Scanner(System.in);
        //datum einlesen
        System.out.println("Geben sie den Tag ein");
        int tag = scan.nextInt();
        System.out.println("Geben sie den Monat ein");
        int monat = scan.nextInt();
        System.out.println("Geben sie das Jahr ein");
        int jahr = scan.nextInt();

        scan.close();
        //gibt ersten beiden ziffern zurück
        int c = Integer.parseInt("'" + jahr).substring(0, 2);

        if(monat == 1 && jahr % 100 == 0 || monat == 2 && jahr % 100 == 0) {
            c = c - 1;
        }

        if(monat == 1 || monat == 2) {
            jahr = jahr - 1;
            jahr = Integer.parseInt("'" + jahr).substring(2));
        //gibt letzten beiden ziffern zurück
        } else jahr = Integer.parseInt("'" + jahr).substring(2));
        //monate laut julianischer zählung
        switch(monat) {
            case 1 : monat = 11;
            break;
            case 2 : monat = 12;
            break;
            case 3 : monat = 1;
            break;
            case 4 : monat = 2;
            break;
            case 5 : monat = 3;
            break;
            case 6 : monat = 4;
            break;
            case 7 : monat = 5;
            break;
            case 8 : monat = 6;
            break;
            case 9 : monat = 7;
            break;
            case 10 : monat = 8;
            break;
            case 11 : monat = 9;
        }
    }
}

```

```

        break;
    case 12 : monat = 10;
        break;
    }

double w = (tag + (2.6 * monat - 0.2) + jahr + (jahr / 4) + (c / 4) - 2 * c) % 7 ;
int ergebnis = (int)Math.floor(w);
ergebnis = ergebnis % 7;
//berechnung falls ergebnis negativ ist
if(ergebnis < 0) {
    ergebnis = Math.floorDiv(ergebnis, 7);
}
//berechnung des wochentages, negative cases falls negatives ergebnis
switch(ergebnis) {
    case 0: wochentag = "Sonntag";
        break;
    case 1: wochentag = "Montag";
        break;
    case 2: wochentag = "Dienstag";
        break;
    case 3: wochentag = "Mittwoch";
        break;
    case 4: wochentag = "Donnerstag";
        break;
    case 5: wochentag = "Freitag";
        break;
    case 6: wochentag = "Samstag";
        break;
    case -1: wochentag = "Montag";
        break;
    case -2: wochentag = "Dienstag";
        break;
    case -3: wochentag = "Mittwoch";
        break;
    case -4: wochentag = "Donnerstag";
        break;
    case -5: wochentag = "Freitag";
        break;
    case -6: wochentag = "Samstag";
        break;
    default : wochentag = "Irgendwas war falsch";
        break;
}
System.out.println(wochentag);
}

//@author maximilian raspe
import java.util.Scanner;
public class Potenzen {
    public static void main (String[] args) {

```

```

Scanner scan = new Scanner(System.in);
System.out.println("Zahl a eingeben");
double a = scan.nextDouble();
System.out.println("Zahl b eingeben");
double b = scan.nextDouble();
int x = (int) Math.floor(Math.random() * 80) + 1;
potenzieren(a,b);

}

public static void potenzieren(double a, double b) {
    int counter = 0;
    double zwischenergebnis = 0;
    double ergebnis = 1;
    boolean fertig = false;
    while(counter < b - 1) { //prüft ob zähler schon die anzahl der durchläufe der potenz erreicht
hat
        if(zwischenergebnis == 0) {
            zwischenergebnis = multiplizieren(a, a);
            counter++;
        }
        ergebnis = multiplizieren(zwischenergebnis, a);
        zwischenergebnis = ergebnis;
        counter++;
    }
    System.out.println(ergebnis);
}

public static double multiplizieren(double a, double b) {
    double sum = 0;
    if(a > b) { //wenn a > b, dann wird so oft addiert, bis das ergebnis erreicht wurde, das selbe gilt
für die unteren ifs
        for(int i = 0; i < a; i++) {
            sum = sum + b;
        }
        return sum;
    }
    if(b > a) {
        for (int i = 0; i < b; i++) {
            sum = sum + a;
        }
        return sum;
    }
    if(b == a) {
        for (int i = 0; i < b; i++) {
            sum = sum + a;
        }
        return sum;
    }
    return sum;
}
//@author maximilian raspe

```

```

import java.awt.*;

public class Schnee {

    public static boolean[][] schnee = new boolean[79][26];
    public static void main (String[] args) {

        Console.setBackground(Color.black);
        Console.setForeground(Color.white);

        //schnee wird zufällig generiert und dann mit update snow nach unten fallen gelassen
        while(true) {
            int x1 = (int)Math.floor(Math.random() * 78) + 1;
            int x2 = (int)Math.floor(Math.random() * 78) + 1;
            int x3 = (int)Math.floor(Math.random() * 78) + 1;
            int x4 = (int)Math.floor(Math.random() * 78) + 1;
            int x5 = (int)Math.floor(Math.random() * 78) + 1;
            for(int y = 1; y <= 25; y++) {
                // Console.wait(50);
                updateSnow(x1, y);
                // Console.wait(50);
                updateSnow(x2, y);
                // Console.wait(50);
                updateSnow(x3, y);
                // Console.wait(50);
                updateSnow(x4, y);
                // Console.wait(50);
                updateSnow(x5, y);
            }
        }
    }

    // lässt den schnee weiter nach unten fallen
    public static void updateSnow(int x, int y) {
        if(y == 25){
            Console.gotoXY(x, y);
            Console.write("#");
            schnee[x][y] = true;
        }
        if(y < 25 && schnee[x][y + 1] == true) {
            Console.gotoXY(x, y - 1);
            Console.write(" ");
            Console.gotoXY(x, y - 1);
            Console.write("#");
            schnee[x][y] = true;
        }
        else if(y < 25 && schnee[x][y + 1] == false) {
            Console.gotoXY(x, y - 1);
            Console.write(" ");
            Console.gotoXY(x, y);
            Console.write("*");
        }
    }
}

```

```

        }
    }

//@author maximilian raspe
import java.util.Scanner;
public class VierGewinnt {

    public static boolean spieler = true; //true = spieler 1, false = spieler 2
    public static char feld[][] = new char[7][6];
    public static boolean fertig = false;

    public static void main(String[] args) {

        int position = 0;

        feldVorbereiten();

        while(true) {
            Scanner scan = new Scanner(System.in);
            System.out.println("Spieler " + spieler + " ist an der Reihe(true = spieler 1, false = spieler 2");
        );

        //programm wird bei einer anderen eingabe als einem int beendet
        if(scan.hasNextInt()) {
            position = scan.nextInt();
        } else System.exit(1);

        //abbruchbedingung, prüfen ob zahl zwischen 0 und 6 liegt und ob die reihe voll ist
        if(position == 0 || position == 1 || position == 2 || position == 3 || position == 4 || position ==
5 || position == 6) {
            int counter = 0;
            for (int j = 0; j < 6; j++) {
                if (feld[position][j] == 'o' || feld[position][j] == 'x') {
                    counter++;
                if (counter >= 6) { //wenn reihe voll ist, solange neu einlesen bis ein nutzbarer wert
rauskommt
                    boolean neuEingeben = false;

                    while (neuEingeben == false) {
                        System.out.println("Bitte neue eingabe");
                        position = scan.nextInt();
                        counter = 0;
                        for (int k = 0; k < 6; k++) {
                            if (feld[position][k] == 'o' || feld[position][k] == 'x') {
                                counter++;
                            }
                        }
                        if(counter < 6) {
                            neuEingeben = true;
                        }
                    }
                }
            }
        }
    }
}

```

```

        }

    }

} else { //falls die zahl ungleich 0-6 sein sollte, solange einlesen bis das der fall ist
    boolean neuEingeben = false;
    while (neuEingeben == false) {
        System.out.println("Bitte neue eingabe");
        position = scan.nextInt();
        int counter = 0;
        for (int k = 0; k < 6 && position <= 6 && position >= 0; k++) {
            if (feld[position][k] == 'o' || feld[position][k] == 'x') {
                counter++;
            }
            if(counter < 6 && position <= 6 && position >= 0) {
                neuEingeben = true;
            }
        }
    }
    setzeStein(spieler, position, feld);
    if(spieler == true) {
        spieler = false;
    } else spieler = true;
}

}

//setzt die steine im array, je nach eingabe und welcher spieler dran ist
private static void setzeStein(boolean spieler1, int position, char [][] field) {
    if(spieler1 == true) {
        for(int i = 0; i < 6; i++) {
            if(field[position][i] == 'x' || field[position][i] == 'o') {
                field[position][i - 1] = 'x';
                break;
            }
            else if(i == 5) {
                field[position][i] = 'x';
            }
        }
    }
    if(spieler1 == false) {
        for(int i = 0; i < 6; i++) {
            if(field[position][i] == 'x' || field[position][i] == 'o') {
                field[position][i - 1] = 'o';
                break;
            }
            else if(i == 5) {
                field[position][i] = 'o';
            }
        }
    }
    maleFeld(field);
}

```

```
}

//malt das feld mithilfe der eingabe
private static void maleFeld(char[][] field) {
    System.out.flush();
    for(int i = 0; i < 6; i++) {
        for(int j = 0; j < 7; j++) {
            System.out.print(field[j][i]);
        }
        System.out.println();
    }
}

//bereitet das feld vor dem spiel vor, damit es wie ein "feld" aussieht
private static void feldVorbereiten() {
    for(int i = 0; i < 6; i++) {
        for(int j = 0; j < 7; j++) {
            feld[j][i] = ' ';
        }
    }
}
```