

1.1) Debugger geht jeden Schritt, den das Programm gehen würde einzeln durch und zeigt immer den derzeitigen Wert aller variablen an. Wenn sich diese verändern wird dies gelb hinterlegt. Der Debugger hilft dann hierbei leichter Fehler zu finden und diese zu korrigieren.

1.2) Fehler ist, dass das j nie erhöht wird und somit j immer <50 bleibt. Dadurch kommt es zu einer Endlosschleife. Außerdem fehlt bei der Raute wenn man das i++ in j++ ändert die letzte Zeile, was sich durch i < 51 beheben lässt, da er so noch die letzte Zeile ausgibt.

2.1) Code Conventions ist das Erstellen von Quellcode nach bestimmten vorgegebenen Regeln. Es gibt vor, wie man ein Programm formal und strukturell gestalten sollte. Es wirken hier drei Aspekte zusammen:

- Die Vorschrift, welche Regeln bzw Standards definiert.
- Die Handlung, welche das Umsetzen dieser Regeln beschreibt.
- Das Ergebnis, welches auf Einhaltung der Vorschriften überprüfbar ist.

2.2) lesbarkeit, verständlichkeit, wartbarkeit

2.3) -Leser können sich auf den Inhalt konzentrieren, statt auf das Layout

- Leser können Code schneller verstehen. Rückschlüsse auf bisherige Erfahrungen bezogen werden können
- erleichtern das kopieren, ändern und pflegen des codes
- man ist oft nie der einzige, der an einem Programm arbeitet

3.1)-1 Deklaration pro Zeile

-Variablen da initialisieren wo sie deklariert sind

-Variablen immer am Blockanfang deklarieren

-kein leerzeichen zwischen einem methodennamen und der klammer

-geschweifte klammer am ende der deklaration

3.2)

```
if (condition) {  
    statements;  
}
```

```
if (condition) {  
    statements;  
} else {  
    statements;  
}
```

```
if (condition) {  
    statements;  
} else if (condition) {  
    statements;  
} else {  
    statements;  
}
```

Die Klammern dürfen nicht weggelassen werden, da das, was zwischen {} steht ausgeführt wird falls der Fall eintritt.

3.3)

- ein wort gefolgt von einer klammer sollte mit einem leerzeichen getrennt werden
- ein leerzeichen nach einem komma in argumenten listen
- alle operatoren außer . Sollten mit einem leerzeichen getrennt werden
- ausdrücke in einem for statement sollten von einem leerzeichen getrennt werden
- nach einem cast soll ein leerzeichen folgen

3.4)

Klassen: -erster buchstabe immer groß

- “UpperCamelCase“
- wenn mehrere worte in einem identifier, anfangsbuchstaben groß schreiben
- für klassen nomen verwenden

Methoden:-erster buchstabe immer klein

- sollten kombination aus verben und nomen darstellen
- bsp: getTasche()

Variablen:- „lowerCamelCase“, erster buchstabe immer klein

3.5) Man Unterscheidet:

- Blockcomments
- Single-Line Comments
- Trailing Comments
- End-Of-Line Comments
- Documentation Comments

4.1) @author

@version
@since
@see
@serial
@serialField
@param
@return
@exception
@throws

```

@deprecated
{@inheritDoc}
{@link reference}
{@linkPlain reference}
{@value}
{@docRoot}
{@code}
{@literal}

4.2) @version erzeugt einen versionseintrag
@param ist die parameterbeschreibung einer methode
@return beschreibt den rückgabewert einer method

//@author maximilian raspe
import java.awt.Color;
import java.util.Arrays;

public class Bundestagswahl {

    public static void main(String[] args) {

        boolean fertig = false;

        //überschrift
        Console.gotoXY(4, 2);
        Console.write("BUNDESTAGSWAHL");

        //zufallszahlen berechnen
        int resultGrüne = (int)(Math.random() * 50) + 1;
        int resultLinke = (int)(Math.random() * 50) + 1;
        int resultAndere = (int)(Math.random() * 50) + 1;
        int resultSPD = (int)(Math.random() * 50) + 1;
        int resultCDU = (int)(Math.random() * 50) + 1;
        int counter = 0; // counter zählt den fortgeschritt

        //namen vor strahlen
        Console.setForeground(java.awt.Color.blue);
        Console.gotoXY( 2, 5 );
        Console.write("CDU");

        Console.setForeground(java.awt.Color.red);
        Console.gotoXY( 2, 7 );
        Console.write("Linke");

        Console.setForeground(java.awt.Color.yellow);
        Console.gotoXY( 2, 9 );
        Console.write("Andere");

        Console.setForeground(java.awt.Color.green);
        Console.gotoXY( 2, 11 );
        Console.write("Grüne");

        Console.setForeground(java.awt.Color.white);
        Console.gotoXY( 2, 13 );
        Console.write("SPD");

        //sortiere array der größe nach
        int[] arr = {resultGrüne, resultLinke, resultAndere, resultSPD, resultCDU};
        Arrays.sort(arr);

        while (fertig == false) {
            for(int i = 8; i < arr[0] + 8; i++) { //schleife für kleinste zahl, geht solange weiter, bis i > zahl an

```

```

index 0 ist

        Console.setForeground(java.awt.Color.blue);
        Console.gotoXY( i + 1, 5 );
        Console.wait(25);
        Console.write("I ");
        Console.gotoXY(70, 5);
        Console.write(counter + "");

        Console.setForeground(Color.red);
        Console.gotoXY( i + 1, 7 );
        Console.wait(25);
        Console.write("I");
        Console.gotoXY(70, 7);
        Console.write(counter + "");

        Console.setForeground(Color.yellow);
        Console.gotoXY( i + 1, 9 );
        Console.wait(25);
        Console.write("I");
        Console.gotoXY(70, 9);
        Console.write(counter + "");

        Console.setForeground(Color.green);
        Console.gotoXY( i + 1, 11 );
        Console.wait(25);
        Console.write("I");
        Console.gotoXY(70, 11);
        Console.write(counter + "");

        Console.setForeground(Color.white);
        Console.gotoXY( i + 1, 13 );
        Console.wait(25);
        Console.write("I");
        Console.gotoXY(70, 13);
        Console.write(counter + "");
        counter++;

    }

for(int i = counter + 8; counter + 8 < arr[1]; i++) { //schleife für nächst größere zahl
    if(resultCDU >= arr[1]) {
        Console.setForeground(java.awt.Color.blue);
        Console.gotoXY(8 + counter + 1, 5 );
        Console.wait(25);
        Console.write("I ");
        Console.gotoXY(70, 5);
        Console.write(counter + "");
    }
    if(resultLinke >= arr[1]) {
        Console.setForeground(java.awt.Color.red);
        Console.gotoXY( 8 +counter + 1, 7 );
        Console.wait(25);
        Console.write("I ");
        Console.gotoXY(70, 7);
        Console.write(counter + "");
    }
    if(resultAndere >= arr[1]) {
        Console.setForeground(java.awt.Color.yellow);
        Console.gotoXY(8 + counter + 1, 9 );
        Console.wait(25);
        Console.write("I ");
        Console.gotoXY(70, 9);
        Console.write(counter + "");
    }
    if(resultGrüne >= arr[1]) {

```

```

        Console.setForeground(java.awt.Color.green);
        Console.gotoXY( 8 + counter + 1, 11 );
        Console.wait(25);
        Console.write("I ");
        Console.gotoXY(70, 11);
        Console.write(counter + "");
    }

    if(resultSPD >= arr[1]) {
        Console.setForeground(java.awt.Color.white);
        Console.gotoXY(8 + counter + 1, 13 );
        Console.wait(25);
        Console.write("I ");
        Console.gotoXY(70, 13);
        Console.write(counter + "");
    }

    counter++;
}

for(int i = counter + 8; counter + 8 < arr[2]; i++) { //schleife für nächst größere zahl
    if(resultCDU >= arr[2]) {
        Console.setForeground(java.awt.Color.blue);
        Console.gotoXY(8 + counter + 1, 5 );
        Console.wait(25);
        Console.write("I ");
        Console.gotoXY(70, 5);
        Console.write(counter + "");
    }

    if(resultLinke >= arr[2]) {
        Console.setForeground(java.awt.Color.red);
        Console.gotoXY(8 + counter + 1, 7 );
        Console.wait(25);
        Console.write("I ");
        Console.gotoXY(70, 7);
        Console.write(counter + "");
    }

    if(resultAndere >= arr[2]) {
        Console.setForeground(java.awt.Color.yellow);
        Console.gotoXY(8 + counter + 1, 9 );
        Console.wait(25);
        Console.write("I ");
        Console.gotoXY(70, 9);
        Console.write(counter + "");
    }

    if(resultGrüne >= arr[2]) {
        Console.setForeground(java.awt.Color.green);
        Console.gotoXY(8 + counter + 1, 11 );
        Console.wait(25);
        Console.write("I ");
        Console.gotoXY(70, 11);
        Console.write(counter + "");
    }

    if(resultSPD >= arr[2]) {
        Console.setForeground(java.awt.Color.white);
        Console.gotoXY( 8 + counter + 1, 13 );
        Console.wait(25);
        Console.write("I ");
        Console.gotoXY(70, 13);
        Console.write(counter + "");
    }

    counter++;
}

for(int i = counter + 8; counter + 8 < arr[3]; i++) { //schleife für nächst größere zahl
    if(resultCDU >= arr[3]) {
        Console.setForeground(java.awt.Color.blue);

```

```

        Console.gotoXY(8 + counter + 1, 5 );
        Console.wait(25);
        Console.write("I ");
        Console.gotoXY(70, 5);
        Console.write(counter + "");

    }

    if(resultLinke >= arr[3]) {
        Console.setForeground(java.awt.Color.red);
        Console.gotoXY(8 + counter + 1, 7 );
        Console.wait(25);
        Console.write("I ");
        Console.gotoXY(70, 7);
        Console.write(counter + "");

    }

    if(resultAndere >= arr[3]) {
        Console.setForeground(java.awt.Color.yellow);
        Console.gotoXY(8 + counter + 1, 9 );
        Console.wait(25);
        Console.write("I ");
        Console.gotoXY(70, 9);
        Console.write(counter + "");

    }

    if(resultGrüne >= arr[3]) {
        Console.setForeground(java.awt.Color.green);
        Console.gotoXY(8 + counter + 1, 11 );
        Console.wait(25);
        Console.write("I ");
        Console.gotoXY(70, 11);
        Console.write(counter + "");

    }

    if(resultSPD >= arr[3]) {
        Console.setForeground(java.awt.Color.white);
        Console.gotoXY(8 + counter + 1, 13 );
        Console.wait(25);
        Console.write("I ");
        Console.gotoXY(70, 13);
        Console.write(counter + "");

    }

    counter++;

}

for(int i = counter + 8; counter + 8 < arr[4]; i++) { //schleife für größte zahl
    if(resultCDU >= arr[4]) {
        Console.setForeground(java.awt.Color.blue);
        Console.gotoXY(8 + counter + 1, 5 );
        Console.wait(25);
        Console.write("I ");
        Console.gotoXY(70, 5);
        Console.write(counter + "");

    }

    if(resultLinke >= arr[4]) {
        Console.setForeground(java.awt.Color.red);
        Console.gotoXY( 8 + counter + 1, 7 );
        Console.wait(25);
        Console.write("I ");
        Console.gotoXY(70, 7);
        Console.write(counter + "");

    }

    if(resultAndere >= arr[4]) {
        Console.setForeground(java.awt.Color.yellow);
        Console.gotoXY(8 + counter + 1, 9 );
        Console.wait(25);
        Console.write("I ");
        Console.gotoXY(70, 9);
        Console.write(counter + "");

    }
}

```

```

        }
        if(resultGrüne >= arr[4]) {
            Console.setForeground(java.awt.Color.green);
            Console.gotoXY(8 + counter + 1, 11 );
            Console.wait(25);
            Console.write("I ");
            Console.gotoXY(70, 11);
            Console.write(counter + "");
        }
        if(resultSPD >= arr[4]) {
            Console.setForeground(java.awt.Color.white);
            Console.gotoXY(8 + counter + 1, 13 );
            Console.wait(25);
            Console.write("I ");
            Console.gotoXY(70, 13);
            Console.write(counter + "");
        }
        counter++;
    }
    fertig = true;
}

}

if(arr[4] == resultCDU) {
    Console.gotoXY(5, 20);
    Console.write("Gewinner ist CDU");
}
if(arr[4] == resultLinke) {
    Console.gotoXY(5, 20);
    Console.write("Gewinner ist Linke");
}
if(arr[4] == resultAndere) {
    Console.gotoXY(5, 20);
    Console.write("Gewinner ist Andere");
}
if(arr[4] == resultGrüne) {
    Console.gotoXY(5, 20);
    Console.write("Gewinner ist Grüne");
}
if(arr[4] == resultSPD) {
    Console.gotoXY(5, 20);
    Console.write("Gewinner ist SPD");
}
}

}

//@author maximilian Raspe
import java.util.Scanner;
public class WallisFaktoren {

    public static void main(String[] args) {

        Scanner scan = new Scanner(System.in);
        double genauigkeit = scan.nextDouble();
        scan.close();

        double result = 1;
        double x = 2;
        double y = 1;

        for(int i = 1; i <=1000000; i++) //schleife für wallis, zuerst zwischenergebnis, dann y+2, dann wieder

```

```

zwischenergebnis, dann x+2
    result = x / y * result;
    y = y + 2;
    result = x / y * result;
    x = x + 2;

    if(Math.PI - genauigkeit < result * 2) { //prüft ob differenz < result * 2 ist
        System.out.println("Faktoren: " + i );
        System.out.println("0,1: 8");
        System.out.println("0,01: 78");
        System.out.println("0,001 : 785 ");
        System.out.println("0,0001 : 7854 ");
        System.out.println("0,00001 : 78540 ");
        break;
    }
}

System.out.println("derzeitiges ergebnis : " + result * 2);
System.out.println("pi " + Math.PI);
}

}

//@author Maximilian Raspe
public class Wallis {

    public static void main(String[] args) {

        double result = 1;
        double x = 2;
        double y = 1;
        for(int i = 1; i <=1000000; i++) {
            result = x / y * result;
            y = y + 2;
            result = x / y * result;
            x = x + 2;
            if(i% 100000 == 1) {
                System.out.println(result * 2);
            }
        }
        System.out.println(result * 2);
    }
}

//@author Maximilian Raspe
public class Pi {

    public static void main(String[] args) {

        double x;
        double y;
        double counter = 0;
        double durchläufe = 0;
        double result;
        for(int i = 0; i <= 1000000; i++) {      //i = anzahl durchläufe

            x = Math.random();      //generiert zufallszahlen
            y = Math.random();
            if(Math.pow(x, 2) + Math.pow(y, 2) <= 1) {  //wenn eine zahl x^2+y^2 <= 1, dann liegt sie im kreis. dann
zähler +1
                counter++;
            }
        }
    }
}

```

```
        durchläufe++;           //jeder durchlauf wird gezählt
    }
    result = counter / durchläufe;
    System.out.println(4 *result);
    System.out.println("zum Vergleich : " + Math.PI);

}
}
```