1.1)Der 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)Der 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. Er 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
-Variabeln da intialisieren wo sie deklariert sind
-Variabeln 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 kalmmer sollte mit einem leerzeichen getrennt weden

-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

```
import java.awt.Color;
public class Rennen {
public static void main(String[] args) {
          for ( int Ziellinie = -1; Ziellinie < 25; Ziellinie++) {
                              Console.gotoXY(53, Ziellinie +1);
                              Console.setForeground(Color.white);
                              Console.write("|");
                    }
          Console.gotoXY(25, 6);
          Console.write("Hamilton");
          Console.gotoXY(25, 12);
          Console.write("Vettel");
          Console.gotoXY(25, 18);
          Console.write("Maximilian");
          Console.wait(250);
          int Fahrer1 = 0;
          int Fahrer2 = 0;
          int Fahrer3 = 0;
          int Gewinner = 0;
          while (Gewinner < 2000) {
                    while( Fahrer1 < 50) {
                              int x = (int) Math.floor(Math.random() * 6) + 1;
                              int NewPos= Fahrer1 + x;
                              while(Fahrer1 + 1 < NewPos){
                              Console.gotoXY(Fahrer1 + 1  , 7);
                               Console.setForeground(Color.white);
                               Console.write("#");
                               Fahrer1++;
                                        }
                              Console.gotoXY(Fahrer1, 7);
                              Console.setForeground(Color.white);
                              Console.write("#");
                              Console.wait(30);
                              int z = Fahrer1 * 2;
                              Console.gotoXY(60, 7);
                              Console.write(z + "%");
                              Gewinner = Fahrer1 + Gewinner;
                              if ( z > 100) {
                                        Console.gotoXY(40, 23);
                                        Console.write("Hamilton hat gewonnen");
                                        return;
                              }
                              break;
                              }
                    while( Fahrer2 < 50) {
                              int x = (int) Math.floor(Math.random() * 6) + 1;
                              int NewPos= Fahrer2 + x;
                              while(Fahrer2 + 1 < NewPos){
                               Console.gotoXY(Fahrer2 + 1  , 13);
                               Console.setForeground(Color.white);
                               Console.write("#");
                               Fahrer2++;
                                        }
                              Console.gotoXY(Fahrer2, 13);
                              Console.setForeground(Color.white);
                              Console.write("#");
                              Console.wait(30);
                              int z = Fahrer2 * 2;
                              Console.gotoXY(60, 13);
                              Console.write(z + "%");
                              Gewinner = Fahrer2 + Gewinner;
                              if ( z > 100) {
                                        Console.gotoXY(40, 23);
                                        Console.write("Vettel hat gewonnen");
                                        return;
                              }
                              break;
                              }
                    while( Fahrer3 < 50) {
                              int x = (int) Math.floor(Math.random() * 6) + 1;
                              int NewPos= Fahrer3 + x;
                              while(Fahrer3 + 1 < NewPos){
                              Console.gotoXY(Fahrer3 + 1  , 19);
                               Console.setForeground(Color.white);
                               Console.write("#");
                               Fahrer3++;
                                        }
                              Console.gotoXY(Fahrer3, 19);
                              Console.setForeground(Color.white);
                              Console.write("#");
                              Console.wait(30);
```

```
                                int z = Fahrer3 * 2;
                                Console.gotoXY(60, 19);
                                Console.write(z + "%");
                                Gewinner = Fahrer3 + Gewinner;
                                if ( z > 100) {
                                          Console.gotoXY(40, 23);
                                          Console.write("Maximilian hat gewonnen");
                                          return;
                                }

                                break;
                                }

                        }
                }
}



public class Wallis {
public static void main (String[] args) {
    double PiHalbe = 1;
    for (int zähler = 2; zähler < 1000001; zähler += 2) {
        PiHalbe = PiHalbe * zähler/(zähler-1) * zähler/(zähler+1);

        if(zähler % 100000 == 0)
            System.out.println("Zwischenresultat: PI = " + PiHalbe*2);
    }

    System.out.println("Endresultat: PI = " + PiHalbe*2);
          }
}

public class Pi {
    public static void main(String[] args) {
        int drinnen = 0;
        int imquadrat = 0;

        double radius = 0.5;
        double radiusquadrat = Math.pow(radius,2);

        for(imquadrat = 1; imquadrat < 1000001; imquadrat++) {
            double x = Math.random();
            double y = Math.random();

            if(Math.pow(x-radius,2) + Math.pow(y-radius,2) < radiusquadrat) {
                drinnen++;
            }

            if(imquadrat % 100000 == 0)
               System.out.println("Zwischen: " + (double)4 * drinnen / imquadrat);
        }

        System.out.println("Ende: " + (double)4 * drinnen / imquadrat + " (Math.PI = " + Math.PI + ")");
    }
}
```