Wiederholung

Frohes Fest und Guten Rutsch !

Inhaltsverzeichnis

Programme kompilieren

- Wie lautet der Dateiname dieser Textdatei?
  `Testprogramm.java`
- Wie lautet der Konsolen-Befehl, um dieses Programm zu kompilieren?
  `javac Testprogramm.java`
- Was entsteht durch das Kompilieren?
  `Testprogramm.class`
- Wie lautet der Konsolen-Befehl, um das Programm auszuführen?
  `java Testprogramm`

Eclipse bedienen

Projekt anlegen (beliebiger Name)

Paket anlegen ist optional

Klasse anlegen (Testprogramm), darin befindet sich der Programmtext

Rechnen mit Literalen

```
6+6*2
= 6+12
= 18

1%2 + 2%1
= 1 + 0
= 1

('a' > 'b') || ('A' < 'B')
= false || true
= true

true != (!false && true)
= true != (true && true)
= true != true
= false

(5 >= 7-9) && !(8%5 <= 3)
= (5 >= -2) && !(3 <= 3)
= true && !true
= true && false
= false

(false || (0 == 0)) && (2 == (!true ? 0 : 1))
= (false || true) && (2 == (false ? 0 : 1))
= true && (2 == 1)
= true && false
= false

(2.2 + 1.1 < 0.5*3.3) ? 1.5 : 2.5
= (3.3 < 1.65) ? 1.5 : 2.5
= false ? 1.5 : 2.5
= 2.5
```

Rechnen mit Variablen

```
(x*y)%(x+y)
= (7*4)%(7+4)
= 28%11
= 6

(a || b) && (x > z)
= (true || false) && (7 > 1.5)
= (true || false) && (7.0 > 1.5)
= true && true
= true

(b ? x : y) - z
= (false ? 7 : 4) - 1.5
= 4 - 1.5
= 4.0 - 1.5
= 2.5

(a && true) || (a && b)
= (true && true) || (true && false)
= true || false
= true

(9 > y) != (!a != !b)
= (9 > 4) != (!true != !false)
= true != (false != true)
= true != true
= false

(x = y)*(z = x) < ((a == b) ? 10 : 15) + 5
= (x = 4)*(z = x) < ((a == b) ? 10 : 15) + 5
= 4*(z = 4) < ((a == b) ? 10 : 15) + 5
= 4*4 < ((true == false) ? 10 : 15) + 5
= 16 < (false ? 10 : 15) + 5
= 16 < 15 + 5
= 16 < 20
= true
```

Sequenzielle Anweisungen

```java
import java.util.Scanner;

public class Miles {
	public static void main(String[] args) {
		Scanner scanner = new Scanner(System.in);

		System.out.print("Meilen: ");
		double mi = scanner.nextDouble();

		double km = mi/0.62137;
		System.out.println("Kilometer: "+km);

		scanner.close();
	}
}
```

Selektive Anweisungen

```java
public class Weekday {
	public static void main(String[] args) {

		int r = (int) (7 * Math.random());
		String day;

		switch (r) {
		case 0:
			day = "Montag";
			break;
		case 1:
			day = "Dienstag";
			break;
		case 2:
			day = "Mittwoch";
			break;
		case 3:
			day = "Donnerstag";
			break;
		case 4:
			day = "Freitag";
			break;
		case 5:
			day = "Samstag";
			break;
		case 6:
			day = "Sonntag";
			break;
		default:
			day = "ERROR";
			break;
		}

		System.out.println(day);

	}
}
```

Iterative Anweisungen

```
import java.util.Scanner;

public class ChristmasTree {
	public static void main(String[] args) {
		Scanner scanner = new Scanner(System.in);

		System.out.print("Größe: ");
		int h = scanner.nextInt();

		for (int i=0; i<h; ++i) {
			System.out.print(' ');
		}
		System.out.println('*');

		for (int r = 1; r <= h; ++r) {
			for (int i = 0; i < h-r; ++i) {
				System.out.print(' ');
			}
			System.out.print('/');
			for (int i = 0; i < 2*r-1; ++i) {
				System.out.print(' ');
			}
			System.out.println('\\');
		}

		for (int i=0; i<=2*h; ++i) {
			System.out.print('-');
		}
		System.out.println();

		for (int i=0; i<h; ++i) {
			System.out.print(' ');
		}
		System.out.println('|');

		scanner.close();
	}
}
```

Methoden schreiben A

```
public class Methods {

	public static int[] digitArray(long nr) {

		int[] array = new int[1+(int)Math.log10(nr)];

		for (int i=array.length-1; nr>0; --i) {
			array[i] = (int)(nr%10);
			nr = nr/10;
		}

		return array;

	}
}
```

Methoden schreiben B

```
public class Methods {

	public static boolean[] bitArray(byte nr) {

		boolean[] array = new boolean[8];

		for (int i=7; i>=0; --i) {
			array[i] = (1&nr) > 0;
			nr = (byte)(nr >> 1);
		}

		return array;

	}
}
```

Rekursion

```
public class Hanoi {

	public static void move(int height, char from, char to, char via) {
		if (height > 0) {
			move(height-1,from,via,to);
			System.out.println("Verschiebe Scheibe " + height + " von " + from
+ " nach " + to);
			move(height-1,via,to,from);
		}
	}

	public static void main (String[] args) {
		move(3,'A','B','C');
	}

}
```

Eigene Klassen schreiben

```
public class Point {

	private double x;
	private double y;

	public Point() {
		this.x = 0;
		this.y = 0;
	}

	public Point(double x, double y) {
		this.x = x;
		this.y = y;
	}

	public double getX() {
		return this.x;
	}
	public double getY() {
		return this.y;
	}

	public double distanceTo(Point other) {
		double dx = this.x - other.x;
		double dy = this.y - other.y;
		return Math.hypot(dx,dy);
	}

}


public class Triangle {

	private final double EPS = 0.000001;

	private Point a;
	private Point b;
	private Point c;

	public Triangle() {
		this.a = new Point(1,0);
		this.b = new Point(0,0);
		this.c = new Point(0,1);
	}

	public Triangle(Point a, Point b, Point c) {
		this.a = a;
		this.b = b;
		this.c = c;
	}

	public double area() {
		double a = this.a.distanceTo(this.b);
		double b = this.b.distanceTo(this.c);
		double c = this.c.distanceTo(this.a);
		double s = (a+b+c)/2;
		return Math.sqrt(s*(s-a)*(s-b)*(s-c));
	}

	public boolean contains(Point p) {
		Triangle t1 = new Triangle(this.a,this.b,p);
		Triangle t2 = new Triangle(this.a,p,this.c);
		Triangle t3 = new Triangle(p,this.b,this.c);
		return t1.area() + t2.area() + t3.area() < this.area() + EPS;
	}

}


public class Test {
	public static void main(String[] args) {
		Triangle t = new Triangle();
		Point p = new Point(0.5,0.5);
		System.out.println(t.contains(p));
	}
}
```