

May 11 2010 08:35

TransportUnternehmenTest2.java

Page 1000000000

```

1 //=====
2 //      * Letsch Informatik *      www.LetschInfo.ch      CH-8636 Wald
3 //      Beratung, Ausbildung und Realisation in Software-Engineering
4 //=====
5 // Project   : Master of Advanced Studies in Software-Engineering 2010
6 // Modul     : OO-Einführung mit UML und Java
7 //           : Teil: UML -> Java
8 // Title     : Übung 6: Vererbung+Polymorphismus und Date-Klasse
9 // Author    : Thomas Letsch
10 // Tab-Width : 2
11 // *///=====
12
13 * Description:
14 Klassen-Hierarchie mit "Fahrzeug" als Basis-Klasse.
15 Klasse "Strassenfahrzeug" und "Wasserfahrzeug" davon abgeleitet.
16 Strassenfahrzeug zusätzlich mit zurückgelegten Kilometer.
17 Wasserfahrzeug zusätzlich mit zurückgelegten Seemeilen.
18 Transport-Unternehmen mit mehreren Fahrzeugen.
19 Fahrzeug mit Datum der ersten Inverkehrsetzung.
20
21 * History   :
22 09.05.1999: Initial-Version
23 10.05.2000: WIN32
24 14.05.2001: Überall Elementinitialisierungsliste.
25             WIN32-spezielle Anpassungen gelöscht.
26 25.04.2002: Von C++ auf Java portiert.
27 14.05.2003: Neue File-Aufteilung betref Eclipse.
28 14.05.2004: Kosmetik.
29 16.05.2005: TransportUnternehmenTest::printInfos() -> printlnInfos.
30 15.05.2006: 'ErsteInverkehrsetzung' von Übung 7 in Übung 6.
31 CVS: $Revision: 1.16 $ $Date: 2010/05/11 08:40:07 $
32 // *///=====
33 //      1      2      3      4      5      6      7      8
34 //34567890123456789012345678901234567890123456789012345678901234567890
35 //=====
36
37 import java.text.DateFormat;
38 import java.text.ParseException;
39 import java.util.Locale;
40 import java.util.Vector;
41
42 public class TransportUnternehmenTest2 {
43
44     public static void main(String[] args) throws ParseException {
45
46         DateFormat df = DateFormat.getDateInstance(DateFormat.MEDIUM, Locale.GERMAN);
47
48         TransportUnternehmen transUnt = new TransportUnternehmen();
49         transUnt.neuesFahrzeug(new Strassenfahrzeug( 3,  4, 0,
50             df.parse("01.02.1995")));
51         transUnt.neuesFahrzeug(new Strassenfahrzeug(-3, -4, 0,
52             df.parse("03.04.1997")));
53         transUnt.neuesFahrzeug(new Wasserfahrzeug( 9, 12, 0,
54             df.parse("05.06.1996")));
55         transUnt.neuesFahrzeug(new Wasserfahrzeug(-9, 12, 0,
56             df.parse("07.08.1998")));
57         transUnt.printlnInfos();
58         transUnt.allesFahren(0, 0);    // alle kommen nach hause ;- )
59         transUnt.printlnInfos();
60         String datum = "03.04.1997";
61         System.out.println("Fahrzeuge aelter als " + datum + ":");
62         Vector<Fahrzeug> vec = transUnt.getFzAelterAls(df.parse(datum));
63         for (int i = 0; i < vec.size(); i++) {
64             vec.elementAt(i).printlnInfos(datum + " > ");
65         }
66     }
67 }
68
69 }

```

May 11 2010 08:35

TransportUnternehmenTest2.java

Page 2000000000

```

70
71
72
73 /* Session-Log:
74
75 TransportUnternehmen.printInfos():
76 Fz 1: Koord = 3.0/4.0    01.02.1995    KM = 0.0
77 Fz 2: Koord = -3.0/-4.0  03.04.1997    KM = 0.0
78 Fz 3: Koord = 9.0/12.0   05.06.1996    SM = 0.0
79 Fz 4: Koord = -9.0/12.0  07.08.1998    SM = 0.0
80 TransportUnternehmen.allesFahren(): 0.0/0.0
81 Fahrzeug.fahren(): 5.0
82 Strassenfahrzeug.fahren(): total KM: 5.0
83 Fahrzeug.fahren(): 5.0
84 Strassenfahrzeug.fahren(): total KM: 5.0
85 Fahrzeug.fahren(): 15.0
86 Wasserfahrzeug.fahren(): total SM: 8.099352051835853
87 Fahrzeug.fahren(): 15.0
88 Wasserfahrzeug.fahren(): total SM: 8.099352051835853
89 TransportUnternehmen.printInfos():
90 Fz 1: Koord = 0.0/0.0    01.02.1995    KM = 5.0
91 Fz 2: Koord = 0.0/0.0    03.04.1997    KM = 5.0
92 Fz 3: Koord = 0.0/0.0    05.06.1996    SM = 8.099352051835853
93 Fz 4: Koord = 0.0/0.0    07.08.1998    SM = 8.099352051835853
94 Fahrzeuge aelter als 03.04.1997:
95 03.04.1997 > : Koord = 0.0/0.0    01.02.1995    KM = 5.0
96 03.04.1997 > : Koord = 0.0/0.0    05.06.1996    SM = 8.099352051835853
97
98 */

```

May 11 2010 08:18

TransportUnternehmen.java

Page 1

```

1 // MAS-SE-10: Übung 6: Vererbung+Polymorphismus und Date-Klasse
2
3 import java.util.Date;
4 import java.util.Vector;
5
6 class TransportUnternehmen {
7
8     private final static int MAX_FAHRZEUGE = 10;
9
10    private Fahrzeug[] mFahrzeuge;
11    private int mAnzahlFahrzeuge;
12
13
14    public TransportUnternehmen() {
15        mFahrzeuge = new Fahrzeug[MAX_FAHRZEUGE];
16    }
17
18
19    public boolean neuesFahrzeug(Fahrzeug pFz) {
20        if (mAnzahlFahrzeuge < MAX_FAHRZEUGE) {
21            mFahrzeuge[mAnzahlFahrzeuge] = pFz;
22            mAnzahlFahrzeuge++;
23            return true;
24        }
25        else {
26            return false; // kein Platz mehr für zusätzliches Fahrzeug
27        }
28    }
29
30
31    public void allesFahren(double pX, double pY) {
32        System.out.println("TransportUnternehmen.allesFahren(): " + pX + "/" + pY);
33        for (int i = 0; i < mAnzahlFahrzeuge; i++) {
34            mFahrzeuge[i].fahren(pX, pY);
35            // Zum Kompilationszeitpunkt nicht bestimmt welches fahren()
36            // aufgerufen werden soll (ob Strassenfahrzeug.fahren() oder
37            // Wasserfahrzeug.fahre()).
38            // Wird zur Laufzeit entschieden: -> Polymorphismus !
39        }
40    }
41
42
43    public Vector<Fahrzeug> getFzAelterAls(Date pDatum) {
44        Vector<Fahrzeug> vec = new Vector<Fahrzeug>();
45        for (int i = 0; i < mAnzahlFahrzeuge; i++) {
46            Fahrzeug fz = mFahrzeuge[i];
47            if (fz.getErsteInverkehrsSetzung().before(pDatum)) {
48                vec.addElement(fz);
49            }
50        }
51        return vec;
52    }
53
54
55    public void printlnInfos() {
56        System.out.println("TransportUnternehmen.printlnInfos():");
57        for (int i = 0; i < mAnzahlFahrzeuge; i++) {
58            mFahrzeuge[i].printlnInfos("Fz " + (i+1));
59        }
60    }
61
62 }

```

May 11 2010 08:18

Fahrzeug.java

Page 1

```

1 // MAS-SE-10: Übung 6: Vererbung+Polymorphismus und Date-Klasse
2
3 import java.text.DateFormat;
4 import java.util.Date;
5 import java.util.Locale;
6
7 class Fahrzeug {
8
9     private double mPosX;
10    private double mPosY;
11    private Date mErsteInverkehrsSetzung;
12
13
14    public Fahrzeug(double pXinit, double pYinit, Date pErsteInverkehrsSetzung) {
15        mPosX = pXinit;
16        mPosY = pYinit;
17        mErsteInverkehrsSetzung = pErsteInverkehrsSetzung;
18    }
19
20
21    public double fahren(double pX, double pY) {
22        System.out.print("Fahrzeug.fahren(): ");
23        double xDiff = mPosX - pX;
24        double yDiff = mPosY - pY;
25        double km = Math.sqrt(xDiff * xDiff + yDiff * yDiff);
26        mPosX = pX;
27        mPosY = pY;
28        System.out.println(km);
29        return km;
30    }
31
32
33    protected void printInfos(String pPrefix) {
34        System.out.print(pPrefix + ": Koord = " + mPosX + "/" + mPosY + "\t"
35            + DateFormat.getDateInstance(DateFormat.MEDIUM, Locale.GERMAN)
36            .format(mErsteInverkehrsSetzung));
37    }
38
39
40    public void printlnInfos(String pPrefix) {
41        printInfos(pPrefix);
42        System.out.print("\n");
43    }
44
45
46    public double getPosX() {
47        return mPosX;
48    }
49
50
51    public double getPosY() {
52        return mPosY;
53    }
54
55
56    public Date getErsteInverkehrsSetzung() {
57        return mErsteInverkehrsSetzung;
58    }
59
60 }

```

May 11 2010 08:18

Strassenfahrzeug.java

Page 1

```

1 // MAS-SE-10: Übung 6: Vererbung+Polymorphismus und Date-Klasse
2
3 import java.util.Date;
4
5 class Strassenfahrzeug extends Fahrzeug {
6
7     private double mKM; // gefahrene Kilometer
8
9
10    public Strassenfahrzeug(double pXinit, double pYinit, double pKMinut,
11                           Date pErsteInverkehrsSetzung) {
12        // Aufruf des Konstruktors der Basis-Klasse:
13        super(pXinit, pYinit, pErsteInverkehrsSetzung);
14        mKM = pKMinut;
15    }
16
17
18    public double fahren(double pX, double pY) {
19        double km = super.fahren(pX, pY);
20        mKM = mKM + km;
21        System.out.println("Strassenfahrzeug.fahren(): total KM: " + mKM);
22        return km;
23    }
24
25
26    public void printlnInfos(String pPrefix) {
27        printlnInfos(pPrefix);
28        System.out.println("\tKM = " + mKM);
29    }
30
31 }

```

May 11 2010 08:18

Wasserfahrzeug.java

Page 1

```

1 // MAS-SE-10: Übung 6: Vererbung+Polymorphismus und Date-Klasse
2
3 import java.util.Date;
4
5 class Wasserfahrzeug extends Fahrzeug {
6
7     private double mSM; // gefahrene Seemeilen
8
9
10    public Wasserfahrzeug(double pXinit, double pYinit, double pSMinut,
11                           Date pErsteInverkehrsSetzung) {
12        // Aufruf des Konstruktors der Basis-Klasse:
13        super(pXinit, pYinit, pErsteInverkehrsSetzung);
14        mSM = pSMinut;
15    }
16
17
18    public double fahren(double pX, double pY) {
19        double km = super.fahren(pX, pY);
20        mSM = mSM + (km / 1.852);
21        System.out.println("Wasserfahrzeug.fahren(): total SM: " + mSM);
22        return km;
23    }
24
25
26    public void printlnInfos(String pPrefix) {
27        printlnInfos(pPrefix);
28        System.out.println("\tSM = " + mSM);
29    }
30
31 }

```