

Apr 02, 12 11:30

OOPExercise.txt

Page 1/4

```

1:/*****
2: * Compilation: javac Vector.java
3: * Execution: java Vector
4: *
5: * Booksite: http://www.cs.princeton.edu/introcs/34nbody/Vector.java
6: * Implementation of a vector of real numbers.
7: *
8: * This class is implemented to be immutable: once the client program
9: * initializes a Vector, it cannot change any of its fields
10: * (N or data[i]) either directly or indirectly. Immutability is a
11: * very desirable feature of a data type.
12: *
13: * % java Vector
14: * x      = ( 1.0 2.0 3.0 4.0 )
15: * y      = ( 5.0 2.0 4.0 1.0 )
16: * x + y  = ( 6.0 4.0 7.0 5.0 )
17: * 10x    = ( 10.0 20.0 30.0 40.0 )
18: * |x|    = 5.477225575051661
19: * <x, y>  = 25.0
20: * |x - y| = 5.0990195135927845
21: *
22: *
23: * Note that Vector is also the name of an unrelated Java library class.
24: *
25: *****/
26:
27: public final class Vector {
28:     private final int N;           // length of the vector
29:     private final double[] data;   // array of vector's components
30:
31:
32:     // create the zero vector of length n
33:     public Vector(int N) {
34:         this.N = N;
35:         this.data = new double[N];
36:     }
37:
38:
39:     // create a vector from the array d
40:     public Vector(double[] d) {
41:         // Something is missing here . . .
42:
43:
44:         // We need a defensive copy so client can't alter our copy of data[]
45:         // This isn't it!
46:         double[] data = d;
47:
48:
49:
50:
51:
52:
53:
54:
55:
56:
57:     }
58:
59:
60:     // return a + b
61:     public Vector plus(Vector b) {
62:         Vector a = this;
63:         if (a.N != b.N) {throw new RuntimeException("Dimensions disagree"); }
64:         Vector c = new Vector(N);
65:         for (int i = 0; i < N; i++) {
66:             c.data[i] = a.data[i] + b.data[i];
67:         }
68:         return c;
69:     }

```

Apr 02, 12 11:30

OOPExercise.txt

Page 2/4

```

70:
71:     // return a - b
72:     public Vector minus(Vector b) {
73:         Vector a = this;
74:         if (a.N != b.N) {throw new RuntimeException("Dimensions disagree"); }
75:         Vector c = new Vector(N);
76:         for (int i = 0; i < N; i++) {
77:             c.data[i] = a.data[i] - b.data[i];
78:         }
79:         return c;
80:     }
81:
82:
83:     // create and return a new object whose value is (this * factor)
84:     public Vector times(double factor) {
85:         Vector c = new Vector(N);
86:         for (int i = 0; i < N; i++) {
87:             c.data[i] = factor * data[i];
88:         }
89:         return c;
90:     }
91:
92:
93:     // return the corresponding unit vector
94:     public Vector direction() {
95:         Vector a = this;
96:         return a.times(1.0 / a.magnitude());
97:     }
98:
99:     // return the inner product of this Vector a and b
100:    public double dot(Vector b) {
101:
102:
103:
104:
105:
106:
107:
108:
109:
110:
111:
112:
113:
114:
115:
116:     }
117:
118:     // return the Euclidean norm of this Vector a
119:     public double magnitude() {
120:         Vector a = this;
121:         return Math.sqrt(a.dot(a));
122:     }
123:
124:
125:     // return the corresponding coordinate
126:     public double cartesian(int i) {
127:         return data[i];
128:     }
129:
130:     // return a string representation of the vector
131:     public String toString() {
132:
133:
134:
135:
136:
137:
138:     }

```

Apr 02, 12 11:30

OOPExercise.txt

Page 3/4

```
139:
140:
141:
142: // test client
143: public static void main(String[] args) {
144:     double[] xdata = { 1.0, 2.0, 3.0, 4.0 };
145:     double[] ydata = { 5.0, 2.0, 4.0, 1.0 };
146:
147:     Vector x = new Vector(xdata);
148:     Vector y = new Vector(ydata);
149:
150:     System.out.println("x      = " + x);
151:     System.out.println("y      = " + y);
152:     System.out.println("x + y  = " + x.plus(y));
153:     System.out.println("10x   = " + x.times(10.0));
154:     System.out.println("|x|   = " + x.magnitude());
155:     System.out.println("<x, y> = " + x.dot(y));
156: }
157: }
158: }
```

Apr 02, 12 11:30

OOPExercise.txt

Page 4/4

Recommended Book Exercises: 3.3.6, 3.3.7