

```

1  /*
2  * Programmer:  Chad Philip Johnson
3  * Date Created:  Thursday, November 15th, 2012
4  * Date of Last Modification:  Thursday, November 15th, 2012
5  *
6  * Description:
7  * Animal.class contains subroutines for managing local variables for instances of type Animal.
8  */
9
10 import java.util.*;
11 import java.io.Serializable;
12
13 /**
14 * Animal.class contains subroutines for managing local variables for instances of type Animal.
15 *
16 * @author Chad Philip Johnson
17 * @version 1.0
18 */
19
20 public class Animal implements Serializable {
21
22     String strAnimalType;
23     String strAnimalProduce;
24     /* NOTE:  Produce values are not used in objects of this class, which deviates
25     slightly from the requested directions from the assignment for the Animal class;
26     instead, the values of each kind of produce change slightly every week, which is
27     more efficiently handled in the class Farm */
28     double dblProduceValue;    // unused/deprecated
29     int intAnimalFeedNeededToProduce;
30     int intFeedGiven;
31
32     String strSellAnimalProduce;
33     int intMaxFeedPerDay, intLastFeedDay;
34     int intAnimalFeedNeededFullGrown;
35     boolean blnProduceAvailable, blnMature, blnProducePreparedToSell, blnAnimalPreparedToSell;
36
37     /**
38     * Default constructor:
39     * Initialize variables for a new instance of Animal.class.
40     */
41
42     public Animal() {
43
44         this.strAnimalType           = "cow";
45         this.strAnimalProduce        = "milk";
46         this.dblProduceValue          = 1.00;    // unused/deprecated (see note at top)
47         this.intAnimalFeedNeededToProduce = 50;
48         this.intFeedGiven             = 0;
49
50         this.strSellAnimalProduce     = "Beef";

```

```

51     this.intAnimalFeedNeededFullGrown    = 1000;
52     this.intMaxFeedPerDay                = 10;
53     this.intLastFeedDay                  = -1;
54     this.blnMature                       = false;
55     this.blnProduceAvailable             = false;
56     this.blnProducePreparedToSell       = false;
57     this.blnAnimalPreparedToSell        = false;
58
59 }
60
61 /**
62  * Overloaded constructor:
63  * Initialize variables for a new instance of Animal.class and establish different values for kind of animal, animal's kind
64  * of produce, and the
65  * amount of feed required for the animal to produce.
66  * @param strAnimalType The kind of animal.
67  * @param strAnimalProduce The kind of produce for this animal.
68  * @param intAnimalFeedNeededToProduce The amount of feed needed for the animal to produce marketable goods.
69  */
70 public Animal( String strAnimalType, String strAnimalProduce, int intAnimalFeedNeededToProduce ) {
71
72     this.strAnimalType                    = strAnimalType;
73     this.strAnimalProduce                 = strAnimalProduce;
74     this.dblProduceValue                   = 1.00;      // unused/depracated (see note at top)
75     this.intAnimalFeedNeededToProduce     = 50;
76     this.intFeedGiven                     = 0;
77
78     this.strSellAnimalProduce              = "Beef";
79     this.intAnimalFeedNeededFullGrown     = 1000;
80     this.intMaxFeedPerDay                  = 10;
81     this.intLastFeedDay                    = -1;
82     this.blnMature                         = false;
83     this.blnProduceAvailable              = false;
84     this.blnProducePreparedToSell         = false;
85     this.blnAnimalPreparedToSell         = false;
86
87 }
88
89 /**
90  * Overloaded constructor:
91  * Initialize variables for a new instance of Animal.class and establish different values for kind of animal, animal's kind
92  * of produce, the kind of
93  * produce/goods created when the animal is sold, the amount of feed required for the animal to produce, the amount of feed
94  * required for the animal
95  * grow to maturity (and begin producing), and the maximum amount of feed that can be given to the animal in one day.
96  * @param strAnimalType The kind of animal.
97  * @param strAnimalProduce The kind of produce for this animal.
98  * @param strSellAnimalProduce The kind of produce obtained when the animal is sold.
99  * @param intAnimalFeedNeededToProduce The amount of feed needed for the animal to produce marketable goods.

```

```

98     * @param intAnimalFeedNeededFullGrown The amount of feed needed for the animal to reach maturity.
99     * @param intMaxFeedPerDay The maximum amount of feed that can be given to this animal in one day.
100    */
101
102    public Animal( String strAnimalType, String strAnimalProduce, String strSellAnimalProduce, int intAnimalFeedNeededToProduce,
103    int intAnimalFeedNeededFullGrown, int intMaxFeedPerDay ) {
104
105        this.strAnimalType           = strAnimalType;
106        this.strAnimalProduce        = strAnimalProduce;
107        this.strSellAnimalProduce     = strSellAnimalProduce;
108        this.intAnimalFeedNeededToProduce = intAnimalFeedNeededToProduce;
109        this.intAnimalFeedNeededFullGrown = intAnimalFeedNeededFullGrown;
110        this.intMaxFeedPerDay         = intMaxFeedPerDay;
111        this.intLastFeedDay           = -1;
112        this.intFeedGiven              = 0;
113        this.blnMature                 = false;
114        this.blnProduceAvailable       = false;
115        this.blnProducePreparedToSell  = false;
116        this.blnAnimalPreparedToSell   = false;
117
118        this.dblProduceValue           = 1.00;    // unused/depracated (see not at top)
119    }
120
121    /**
122     * Find the number of units of produce available from this animal.
123     *
124     * @return The number of units of produce available for sale.
125     */
126
127    public int unitsOfProduceAvailable() {
128
129        if( intAnimalFeedNeededToProduce > 0 ) {
130
131            return (intFeedGiven / intAnimalFeedNeededToProduce);
132
133        } else {
134
135            return -1;
136
137        }
138
139    }
140
141    /**
142     * Determine the number of units of produce sold and decrement the amount of feed given to the current instance of
143     * Animal.class according to the ratio
144     * of amount of feed per unit of produce.
145     *
146     * @param intNumberOfUnits The number of units of produce to be sold.

```

```

146     */
147
148     public void unitsOfProduceSold( int intNumberOfUnits)  {
149
150         intFeedGiven -= (intNumberOfUnits * intAnimalFeedNeededToProduce);
151
152         if( intFeedGiven < intAnimalFeedNeededToProduce )  {   blnProduceAvailable   = false;   }
153
154     }
155
156     /**
157     * Give feed to the current instance of Animal.class.
158     *
159     * @param intFeedPortion The amount of feed to be given to the current instance.
160     */
161
162     public void feedAnimal( int intFeedPortion )  {
163
164         intFeedGiven += intFeedPortion;
165
166         if( ( blnMature == false ) && ( intFeedGiven >= intAnimalFeedNeededFullGrown ) )  {
167
168             blnMature = true;
169             intFeedGiven -= intAnimalFeedNeededFullGrown;
170
171         }
172
173         if( ( blnMature == true ) && ( intFeedGiven >= intAnimalFeedNeededToProduce ) ) {
174
175             blnProduceAvailable = true;
176
177         }
178
179     }
180
181     /**
182     * Give feed to the current instance. Pass the current "day" of work on the farm to only allow the player to feed the
183     * animal once per day.
184     *
185     * @param intFeedPortion The amount of feed to be given to the current instance.
186     * @param intCurrentFeedDay The current number of "days" the player has completed.
187     */
188
189     public void feedAnimal( int intFeedPortion, int intCurrentFeedDay ) {
190
191         feedAnimal( intFeedPortion );
192
193         intLastFeedDay = intCurrentFeedDay;
194
195     }

```

```

195
196 /**
197  * Provides the player with details about the current instance.
198  * @return Details about the current instance that contain important information for the player such as amount of feed
199  * given, kind(s) of produce,
200  * kind of animal, etc.
201  */
202 public String toString()    {
203
204     // Initialize string
205     String strAnimalStats    = "";
206
207     // Included notifier that produce is ready to be sold
208     if( ( blnMature ) && ( intFeedGiven >= intAnimalFeedNeededToProduce ) && ( intAnimalFeedNeededToProduce > -1 ) )    {
209         strAnimalStats += "* "; }
210
211     // Concatenate animal type and separator
212     strAnimalStats += strAnimalType + ", ";
213
214     // Concatenate animal produce type and separator
215     if( strAnimalProduce != null ) {    strAnimalStats += strAnimalProduce + ", "; }
216
217     // Concatenate animal produce type when sold and separator
218     if( strSellAnimalProduce != null ) {    strAnimalStats += strSellAnimalProduce; }
219
220     // unused/depracated (see note at top)
221     // Concatenate produce value for animal
222     //if( dblProduceValue > 0 ) {    strAnimalStats += currencyFormatter.format( dblProduceValue01 ) + ", "; }
223
224     // Concatenate amount of feed needed to produce
225     if( ( intAnimalFeedNeededFullGrown > intFeedGiven ) && ( blnMature == false ) ) {
226
227         // Growing animal
228         // Concatenate amount of feed given, divisor and amount of feed needed till full grown
229         strAnimalStats += ", " + Integer.toString( intFeedGiven ) + "/" + Integer.toString( intAnimalFeedNeededFullGrown ) +
230         " lbs of feed";
231
232     } else if( intAnimalFeedNeededToProduce > -1 ) {
233
234         // Mature animal that has produces
235         // Concatenate amount of feed given, divisor and amount of feed needed till produce is available
236         strAnimalStats += ", " + Integer.toString( intFeedGiven ) + "/" + Integer.toString( intAnimalFeedNeededToProduce ) +
237         " lbs of feed";
238
239     }
240
241     // Animal maturity indicator
242     if( blnMature ) {

```

```

241         // Mature animal
242         strAnimalStats += " (M)\n";
243
244     } else {
245
246         // Growing animal
247         strAnimalStats += " (m)\n";
248
249     }
250
251     return strAnimalStats;
252
253 }
254
255 /* Accessor/Mutator methods */
256
257 public void setAnimalType( String strAnimalType ) { this.strAnimalType = strAnimalType; }
258
259 public String getAnimalType() { return strAnimalType; }
260
261 public void setAnimalProduce( String strAnimalProduce ) { this.strAnimalProduce = strAnimalProduce; }
262
263 public String getAnimalProduce() { return strAnimalProduce; }
264
265 // unused/depracated (see note at top)
266 public void setProduceValue( double dblProduceValue ) { this.dblProduceValue = dblProduceValue; }
267
268 // unused/depracated (see note at top)
269 public double getProduceValue() { return dblProduceValue; }
270
271 public void setAnimalFeedNeededToProduce( int intAnimalFeedNeededToProduce ) { this.intAnimalFeedNeededToProduce =
intAnimalFeedNeededToProduce; }
272
273 public int getAnimalFeedNeededToProduce() { return intAnimalFeedNeededToProduce; }
274
275 public void setFeedGiven( int intFeedGiven ) { this.intFeedGiven = intFeedGiven; }
276
277 public int getFeedGiven() { return intFeedGiven; }
278
279 public void setSellAnimalProduce( String strSellAnimalProduce ) { this.strSellAnimalProduce = strSellAnimalProduce; }
280
281 public String getSellAnimalProduce() { return strSellAnimalProduce; }
282
283 public void setAnimalFeedNeededFullGrown( int intAnimalFeedNeededFullGrown ) { this.intAnimalFeedNeededFullGrown =
intAnimalFeedNeededFullGrown; }
284
285 public int getAnimalFeedNeededFullGrown() { return intAnimalFeedNeededFullGrown; }
286
287 public void setMaxFeedPerDay( int intMaxFeedPerDay ) { this.intMaxFeedPerDay = intMaxFeedPerDay; }
288

```

```
289     public int getMaxFeedPerDay() { return intMaxFeedPerDay; }
290
291     public void setLastFeedDay( int intLastFeedDay ) { this.intLastFeedDay = intLastFeedDay; }
292
293     public int getLastFeedDay() { return intLastFeedDay; }
294
295     public void setMature( boolean blnMature ) { this.blnMature = blnMature; }
296
297     public boolean getMature() { return blnMature; }
298
299     public void setProduceAvailable( boolean blnProduceAvailable ) { this.blnProduceAvailable = blnProduceAvailable; }
300
301     public boolean getProduceAvailable() { return blnProduceAvailable; }
302
303     public void setProducePreparedToSell( boolean blnProducePreparedToSell ) { this.blnProducePreparedToSell =
304     blnProducePreparedToSell; }
305
306     public boolean getProducePreparedToSell() { return blnProducePreparedToSell; }
307
308     public void setAnimalPreparedToSell( boolean blnAnimalPreparedToSell ) { this.blnAnimalPreparedToSell =
309     blnAnimalPreparedToSell; }
310
311     public boolean getAnimalPreparedToSell() { return blnAnimalPreparedToSell; }
312 }
```