1 /\*\*  
 2 \* This tester DinosaurTester initializes and reads dinos.  
 3 \* It has an interactive menu to display Herbivore and Carnivore objects.  
 4 \*  
 5 \* main() displays a numbered menu for the user to choose from. They can   
 6 \* view dino list, view dino types, print a specific dino they choose,  
 7 \* print all herbivores, print all carnivores, compare herbivore sizes,  
 8 \* or compare carnivore sizes.  
 9 \* printMainMenu() is a method that displays the options of the main menu.  
 10 \* printDinos() accepts ArrayList<Dinosaur> and String type to display  
 11 \* certain categories of dinosaurs.  
 12 \* compareResults() accepts ArrayList<Dinosaur>, two dinosaurs being  
 13 \* compared, and the result of the comparison. It prints the information  
 14 \* on which dino has the bigger size or if they are the same.  
 15 \* printLogo() prints the ASCII logo at program startup "Welcome To Dino Park"  
 16 \*   
 17 \* @author Stephanie Gremillion  
 18 \* @version 17.0.2  
 19 \* @since 2022/07/14  
 20 \*/  
 21   
 22 import java.util.\*;  
 23 import java.util.ArrayList;  
 24 import java.util.Scanner;  
 25   
 26 public class DinosaurTester {  
 27 public static void main(String[] args) {  
 28 // set variables  
 29 String temp;  
 30 int mainMenu = 0;  
 31 int chooseDino = 0;  
 32 int dinoCompare = 0;  
 33 int compareResult = 0;  
 34 Scanner input = new Scanner(System.in);  
 35 ArrayList<Dinosaur> dino = new ArrayList<Dinosaur>();  
 36   
 37 // setting and initializing default dinos  
 38 dino.add(new CT\_Spinosaurus("95 - 70", "Egypt and Morocco", 18.0, 2, "flat and blade-like", 4000, "fish and maybe other dinosaurs"));  
 39 dino.add(new CT\_Velociraptor("74 - 70", "Mongolia", 1.8, 2, "lots of sharp, pointed teeth", 7, "other animals"));  
 40 dino.add(new CH\_Herrerasaurus("228 - 200", "South and North America", 6.0, 2, "large teeth", 350, "other animals"));  
 41 dino.add(new CH\_Staurikosaurus("228 - 200", "South and North America", 2.1, 2, "serrated, laterally compressed, and caudally curved", 12, "smaller prey"));  
 42 dino.add(new HO\_Agilisaurus("169 - 159", "China", 1.5, 2, "leaf-shaped", 40, "leaves"));  
 43 dino.add(new HO\_Lesothosaurus("213 - 200", "Lesotho", 1.0, 2, "small, grinding teeth", 10, "plant material"));  
 44 dino.add(new HS\_Brachiosaurus("155 - 140", "Algeria, Protugal, Tanzania, and USA", 30.0, 4, "large, spoon-shaped", 400, "plant material"));  
 45 dino.add(new HS\_Bellusaurus("180 - 159", "China", 5.0, 4, "unknown", 0, "plants"));  
 46   
 47 // printing dino park logo  
 48 printLogo();  
 49   
 50 //menu loop until user chooses exit  
 51 do {  
 52 mainMenu = 0;  
 53 System.out.println();  
 54 printMainMenu();  
 55 temp = input.next();  
 56 if((temp.matches("[0-9]+") == false) || (Integer.parseInt(temp) < 1) || (Integer.parseInt(temp) > 9)) { // error if invalid selection  
 57 System.out.println("Please enter a valid selection.\n");  
 58 }  
 59 else {  
 60 mainMenu = Integer.parseInt(temp);  
 61 }  
 62   
 63 switch(mainMenu) {  
 64   
 65 // View Dinosaur List  
 66 case 1:  
 67 System.out.println();  
 68   
 69 printDinos(dino, "List All");  
 70 break;  
 71   
 72 // View Dinosaur Types  
 73 case 2:  
 74 System.out.println();  
 75 printDinos(dino, "Types");  
 76 break;  
 77   
 78 // Print Dinosaur  
 79 case 3:  
 80 System.out.println();  
 81 do{ // repeat until valid selection  
 82 chooseDino = 0;  
 83 System.out.println("Choose a dinosaur: ");  
 84 printDinos(dino, "List All Num");  
 85 temp = input.next();  
 86 if((temp.matches("[0-9]+") == false) || (Integer.parseInt(temp) < 1) || (Integer.parseInt(temp) > 9)) { // error if invalid selection  
 87 System.out.println("Please enter a valid selection.\n");  
 88 }  
 89 else {  
 90 chooseDino = Integer.parseInt(temp);  
 91 }  
 92 } while((chooseDino < 1) || (chooseDino > 9));  
 93   
 94 System.out.println();  
 95 if(chooseDino != 9) {  
 96 switch(chooseDino) { // printing picture before dino  
 97 case 1:  
 98 ((Carnivore)dino.get(chooseDino - 1)).printSpino();  
 99 break;  
100 case 2:  
101 ((Carnivore)dino.get(chooseDino - 1)).printVelo();  
102 break;  
103 case 3:  
104 ((Carnivore)dino.get(chooseDino - 1)).printHerr();  
105 break;  
106 case 4:  
107 ((Carnivore)dino.get(chooseDino - 1)).printStaur();  
108 break;  
109 case 5:  
110 ((Herbivore)dino.get(chooseDino - 1)).printAgil();  
111 break;  
112 case 6:  
113 ((Herbivore)dino.get(chooseDino - 1)).printLes();  
114 break;  
115 case 7:  
116 ((Herbivore)dino.get(chooseDino - 1)).printBrach();  
117 break;  
118 case 8:  
119 ((Herbivore)dino.get(chooseDino - 1)).printBell();  
120 break;  
121 default:  
122 break;  
123 }  
124 System.out.println((dino.get(chooseDino - 1)).toString()); // print dino selected   
125 }   
126 else { // if user chose to print all dinos  
127 System.out.println();  
128 printDinos(dino, "Herbivore");  
129 printDinos(dino, "Carnivore");  
130 }  
131 break;   
132   
133 // Print Herbivores  
134 case 4:  
135 System.out.println();  
136 printDinos(dino, "Herbivore");  
137 break;  
138   
139 // Print Carnivores  
140 case 5:  
141 System.out.println();  
142 printDinos(dino, "Carnivore");  
143 break;  
144   
145 // Compare Herbivores Size  
146 case 6:  
147 // choose first dino to compare  
148 do { // repeat until valid selection  
149 chooseDino = 0;  
150 System.out.println();  
151 System.out.println("Choose a dinosaur to compare: ");  
152 printDinos(dino, "Choose Herb");  
153 temp = input.next();  
154 if((temp.matches("[0-9]+") == false) || (Integer.parseInt(temp) < 1) || (Integer.parseInt(temp) > 4)) { // error if invalid selection  
155 System.out.println("Please enter a valid selection.\n");  
156 }  
157 else {  
158 chooseDino = Integer.parseInt(temp);  
159 }  
160 } while((chooseDino < 1) || (chooseDino > 4));  
161   
162 // choose second dino to compare  
163 do { // repeat until valid selection  
164 dinoCompare = 0;  
165 System.out.println();  
166 System.out.println("Choose a dinosaur to compare: ");  
167 printDinos(dino, "Choose Herb");  
168 temp = input.next();  
169 if((temp.matches("[0-9]+") == false) || (Integer.parseInt(temp) < 1) || (Integer.parseInt(temp) > 4)) { // error if invalid selection  
170 System.out.println("Please enter a valid selection.\n");  
171 }  
172 else {  
173 dinoCompare = Integer.parseInt(temp);  
174 }  
175 } while((dinoCompare < 1) || (dinoCompare > 4));  
176   
177 // compare the two dinos then print  
178 System.out.println();  
179 compareResult = ((Herbivore)dino.get(chooseDino + 3)).compareTo((Herbivore)dino.get(dinoCompare + 3));  
180 compareResults(dino, chooseDino + 3, dinoCompare + 3, compareResult);  
181 break;  
182   
183 // Compare Carnivores Size  
184 case 7:  
185 // choose first dino to compare  
186 do { // repeat until valid selection  
187 chooseDino = 0;  
188 System.out.println();  
189 System.out.println("Choose a dinosaur to compare: ");  
190 printDinos(dino, "Choose Carn");  
191 temp = input.next();  
192 if((temp.matches("[0-9]+") == false) || (Integer.parseInt(temp) < 1) || (Integer.parseInt(temp) > 4)) { // error if invalid selection  
193 System.out.println("Please enter a valid selection.\n");  
194 }  
195 else {  
196 chooseDino = Integer.parseInt(temp);  
197 }  
198 } while((chooseDino < 1) || (chooseDino > 4));  
199   
200 // choose second dino to compare  
201 do { // repeat until valid selection  
202 dinoCompare = 0;  
203 System.out.println();  
204 System.out.println("Choose a dinosaur to compare: ");  
205 printDinos(dino, "Choose Carn");  
206 temp = input.next();  
207 if((temp.matches("[0-9]+") == false) || (Integer.parseInt(temp) < 1) || (Integer.parseInt(temp) > 4)) { // error if invalid selection  
208 System.out.println("Please enter a valid selection.\n");  
209 }  
210 else {  
211 dinoCompare = Integer.parseInt(temp);  
212 }  
213 } while((dinoCompare < 1) || (dinoCompare > 4));  
214   
215 // compare the two dinos then print  
216 System.out.println();  
217 compareResult = ((Carnivore)dino.get(chooseDino - 1)).compareTo((Carnivore)dino.get(dinoCompare - 1));  
218 compareResults(dino, chooseDino - 1, dinoCompare - 1, compareResult);  
219 break;  
220   
221 // Exit  
222 case 8:  
223 break;  
224   
225 default:  
226 break;  
227 }  
228 } while(mainMenu != 8);  
229 }  
230   
231   
232 // method to display main menu  
233 public static void printMainMenu() {  
234 System.out.println("Main Menu");  
235 System.out.println("------------");  
236 System.out.println("1. View Dinosaur List");  
237 System.out.println("2. View Dinosaur Types");  
238 System.out.println("3. Print Dinosaur");  
239 System.out.println("4. Print Herbivores");  
240 System.out.println("5. Print Carnivores");  
241 System.out.println("6. Compare Herbivores Size");  
242 System.out.println("7. Compare Carnivores Size");  
243 System.out.println("8. Exit");  
244 }  
245   
246 // method to print dinos, accepts limitations (both types, one type, or numbered for choosing)  
247 public static void printDinos(ArrayList<Dinosaur> dino, String type) {  
248 // variables  
249 int count = 1;  
250   
251 for(int i = 0; i < 8; i++) { // goes through the list of all dinos  
252 if(type.equals("Types")) { // if printing list of dino types  
253 System.out.println(((dino.get(i)).getType()) + " - " + ((dino.get(i)).getDescription()));  
254 i++;  
255 }  
256 else if(type.equals("List All")) { // if printing list of dinos  
257 System.out.println((dino.get(i)).getName());  
258 }  
259 else if(type.equals("List All Num")) { // if printing list of dinos to choose from  
260 System.out.println(count + ". " + (dino.get(i)).getName());  
261 count++;  
262 }  
263 else if((((dino.get(i)).getDiet()).equals(type))) { // if "type" matches dino type  
264 System.out.println((dino.get(i)).toString());  
265 }  
266 else if(type.equals("Choose Herb") && (((dino.get(i)).getDiet()).equals("Herbivore"))) { // if printing herbivores to choose from  
267 System.out.println(count + ". " + (dino.get(i)).getName());  
268 count++;  
269 }  
270 else if(type.equals("Choose Carn") && (((dino.get(i)).getDiet()).equals("Carnivore"))) { // if printing carnivores to choose from  
271 System.out.println(count + ". " + (dino.get(i)).getName());  
272 count++;  
273 }  
274 }  
275 if(type.equals("List All Num")) { // append to end if listing all to choose  
276 System.out.println("9. Print All");  
277 }  
278 }  
279   
280 // method to print compare result  
281 public static void compareResults(ArrayList<Dinosaur> dino, int dinoOne, int dinoTwo, int result) {  
282 switch(result) {  
283 case -1: // dinoTwo is greater than dinoOne  
284 System.out.println((dino.get(dinoTwo)).getName() + " has a size of " + (dino.get(dinoTwo)).getSize()  
285 + "m which is greather than " + (dino.get(dinoOne)).getName()  
286 + "'s size of " + (dino.get(dinoOne)).getSize() + "m.\n");  
287 break;  
288 case 0: // dinoOne and dinoTwo are equal  
289 System.out.println((dino.get(dinoOne)).getName() + " and " + (dino.get(dinoTwo)).getName()  
290 + " have the same size of " + (dino.get(dinoOne)).getSize() + "m.");  
291 break;  
292 case 1: // dinoOne is greater than dinoTwo  
293 System.out.println((dino.get(dinoOne)).getName() + " has a size of " + (dino.get(dinoOne)).getSize()  
294 + "m which is greather than " + (dino.get(dinoTwo)).getName()  
295 + "'s size of " + (dino.get(dinoTwo)).getSize() + "m.\n");  
296 break;  
297 default:  
298 break;  
299 }  
300 }  
301   
302 // method to display dino park logo  
303 public static void printLogo() {  
304 System.out.println("\_\_ \_\_ \_ \_ \_ \_ ");  
305 System.out.println("\\ \\ / /\_ \_| |\_\_ \_ \_ \_ \_ \_ \_ |\_ \_|\_\_ ");  
306 System.out.println(" \\ \\/\\/ // -\_) / \_/ \_ \\ ' \\ / -\_) | |/ \_ \\ ");  
307 System.out.println(" \\\_/\\\_/ \\\_\_\_|\_\\\_\_\\\_\_\_/\_|\_|\_|\\\_\_\_| |\_|\\\_\_\_/ ");  
308   
309 System.out.println(" .=-.-..-.\_ \_,.---.\_ \_ \_\_ ,---. ,--.-.,-. ");  
310 System.out.println(" \_,..---.\_ /==/\_ /==/ \\ .-.\_ ,-.' , - `. .-`.' ,`..--.' \\ .-.,.---. /==/- |\\ \\ ");  
311 System.out.println("/==/, - \\|==|, ||==|, \\/ /, /==/\_, , - \\ /==/, - \\==\\-/\\ \\ /==/ ` \\|==|\_ `/\_ / ");  
312 System.out.println("|==| \_ \_\\==| ||==|- \\| |==| .=. | |==| \_ .=. /==/-|\_\\ | |==|-, .=., |==| , / ");  
313 System.out.println("|==| .=. |==|- ||==| , | -|==|\_ : ;=: - | |==| , '=',\\==\\, - \\ |==| '=' /==|- .| ");  
314 System.out.println("|==|,| | -|==| ,||==| - \_ |==| , '=' | |==|- '..'/==/ - ,| |==|- , .'|==| \_ , \\ ");  
315 System.out.println("|==| '=' /==|- ||==| /\\ , |\\==\\ - ,\_ / |==|, | /==/- /\\ - \\|==|\_ . ,'./==/ '\\ | ");  
316 System.out.println("|==|-, \_`//==/. //==/, | |- | '.='. - .' /==/ - | \\==\\ \_.\\=\\.-'/==/ /\\ , )==\\ /\\=\\.' ");  
317 System.out.println("`-.`.\_\_\_\_.' `--`-` `--`./ `--` `--`--'' `--`---' `--` `--`-`--`--' `--` ");  
318 }  
319 }

1 /\*\*  
 2 \* This is the Dinosaur interface which is implemented by Carnivore and  
 3 \* Herbivore.  
 4 \*   
 5 \* @author Stephanie Gremillion  
 6 \* @version 17.0.2  
 7 \* @since 2022/07/14  
 8 \*/  
 9   
10 public interface Dinosaur {   
11 // base class method headers  
12 public String getTime();  
13 public void setTime( String time );  
14 public String getLocation();  
15 public void setLocation( String location );  
16 public double getSize();  
17 public void setSize( double size);  
18 public int getLegs();  
19 public void setLegs(int legs );  
20 public String getTeeth();  
21 public void setTeeth( String teeth );  
22 public int getWeight();  
23 public void setWeight( int weight );  
24 public String getFood();  
25 public void setFood( String food );  
26 public String getDiet();   
27   
28 // extended class method headers  
29 public String getType();  
30 public String getDescription();  
31   
32 // subclass class method headers  
33 public String getName();  
34 public String getMeaning();  
35 public String getPerson();  
36 }

1 /\*\*  
 2 \* This is the Carnivore abstract class that implements Dinosaur interface  
 3 \* and Comparable<T> interface. C\_Theropod and C\_Herrerasauridus extend   
 4 \* this class.  
 5 \*   
 6 \* This class:  
 7 \* ...contains the all of the shared instance variables for all of  
 8 \* the Carnivores: diet, time, location, size, legs, teeth, weight, food.  
 9 \* ...has a constructor that initializees all of the above instance   
10 \* variables, also setting diet to the appropriate "Carnivore".  
11 \* ...contains all of the setters and getters for the above instance  
12 \* variables.  
13 \* ...overrides the compareTo() method to compare two dino's sizes.  
14 \* ...contains method headers for its subclasses.  
15 \*   
16 \* @author Stephanie Gremillion  
17 \* @version 17.0.2  
18 \* @since 2022/07/14  
19 \*/  
20   
21 abstract class Carnivore implements Dinosaur, Comparable<Carnivore> {  
22 // variables  
23 private String diet;  
24 private String time;  
25 private String location;  
26 private double size;  
27 private int legs;  
28 private String teeth;  
29 private int weight;  
30 private String food;  
31   
32 // constructor  
33 public Carnivore(String tim, String locate, double siz, int leg, String teet, int weigh, String foo) {  
34 diet = "Carnivore";  
35 time = tim;  
36 location = locate;  
37 size = siz;   
38 legs = leg;  
39 teeth = teet;  
40 weight = weigh;  
41 food = foo;  
42 }  
43   
44 // setters  
45 public void setDiet(String d) { diet = d; }  
46 public void setTime(String ti) { time = ti; }  
47 public void setLocation(String lo) { location = lo; }  
48 public void setSize(double s) { size = s; }  
49 public void setLegs(int le) { legs = le; }  
50 public void setTeeth(String t) { teeth = t; }  
51 public void setWeight(int w) { weight = w; }  
52 public void setFood(String f) { food = f; }  
53   
54 // getters  
55 public String getDiet() { return diet; }  
56 public String getTime() { return time; }  
57 public String getLocation() { return location; }  
58 public double getSize() { return size; }  
59 public int getLegs() { return legs; }  
60 public String getTeeth() { return teeth; }  
61 public int getWeight() { return weight; }  
62 public String getFood() { return food; }  
63   
64 // overriding compareTo() method  
65 public int compareTo(Carnivore dino) {  
66 if(this.size > dino.size) {  
67 return 1;  
68 }  
69 else if (this.size < dino.size) {  
70 return -1;  
71 }  
72 else {  
73 return 0;  
74 }  
75 }  
76   
77 //subclass class method headers  
78 public String getName() { return ""; }  
79 public String getMeaning() { return ""; }  
80 public String getPerson() { return ""; }  
81 public void printSpino() {} ;  
82 public void printVelo() {};  
83 public void printHerr() {};  
84 public void printStaur() {};  
85 }

1 /\*\*  
 2 \* This is the Herbivore abstract class that implements Dinosaur interface  
 3 \* and Comparable<T> interface. H\_Ornithiscian and H\_Sauropod extend   
 4 \* this class.  
 5 \*   
 6 \* This class:  
 7 \* ...contains the all of the shared instance variables for all of  
 8 \* the Herbivores: diet, time, location, size, legs, teeth, weight, food.  
 9 \* ...has a constructor that initializees all of the above instance   
10 \* variables, also setting diet to the appropriate "Herbivores".  
11 \* ...contains all of the setters and getters for the above instance  
12 \* variables.  
13 \* ...overrides the compareTo() method to compare two dino's sizes.  
14 \* ...contains method headers for its subclasses.  
15 \*   
16 \* @author Stephanie Gremillion  
17 \* @version 17.0.2  
18 \* @since 2022/07/14  
19 \*/  
20   
21 abstract class Herbivore implements Dinosaur, Comparable<Herbivore> {  
22 // variables  
23 private String diet;  
24 private String time;  
25 private String location;  
26 private double size;  
27 private int legs;  
28 private String teeth;  
29 private int weight;  
30 private String food;  
31   
32 // constructor  
33 public Herbivore(String tim, String locate, double siz, int leg, String teet, int weigh, String foo) {  
34 diet = "Herbivore";  
35 time = tim;  
36 location = locate;  
37 size = siz;   
38 legs = leg;  
39 teeth = teet;  
40 weight = weigh;  
41 food = foo;  
42 }  
43   
44 // setters  
45 public void setDiet(String d) { diet = d; }  
46 public void setTime(String ti) { time = ti; }  
47 public void setLocation(String lo) { location = lo; }  
48 public void setSize(double s) { size = s; }  
49 public void setLegs(int le) { legs = le; }  
50 public void setTeeth(String t) { teeth = t; }  
51 public void setWeight(int w) { weight = w; }  
52 public void setFood(String f) { food = f; }  
53   
54 // getters  
55 public String getDiet() { return diet; }  
56 public String getTime() { return time; }  
57 public String getLocation() { return location; }  
58 public double getSize() { return size; }  
59 public int getLegs() { return legs; }  
60 public String getTeeth() { return teeth; }  
61 public int getWeight() { return weight; }  
62 public String getFood() { return food; }  
63   
64 // overriding compareTo() method  
65 public int compareTo(Herbivore dino) {  
66 if(this.size > dino.size) {  
67 return 1;  
68 }  
69 else if (this.size < dino.size) {  
70 return -1;  
71 }  
72 else {  
73 return 0;  
74 }  
75 }  
76   
77 //subclass class method headers  
78 public String getName() { return ""; }  
79 public String getMeaning() { return ""; }  
80 public String getPerson() { return ""; }  
81 public void printAgil() {};  
82 public void printLes() {};  
83 public void printBrach() {};  
84 public void printBell() {};  
85 }

1 /\*\*  
 2 \* This is the C\_Theropod concrete class that extend Carnivore.  
 3 \*   
 4 \* This class:  
 5 \* ...has a constructor that sends the following variables to super   
 6 \* (Carnivore): time, location, size, legs, teeth, weight, food.  
 7 \* ...has appropriate getters that return type "Theropod" as well as  
 8 \* returning the appropriate description.  
 9 \* ...overrides toString() to display the base information about this  
10 \* type of dino.  
11 \*   
12 \* @author Stephanie Gremillion  
13 \* @version 17.0.2  
14 \* @since 2022/07/14  
15 \*/  
16   
17 public class C\_Theropod extends Carnivore {  
18 // constructor  
19 public C\_Theropod(String tim, String locate, double siz, int leg, String teet, int weigh, String foo) {  
20 super(tim, locate, siz, leg, teet, weigh, foo);  
21 }  
22   
23 // getters  
24 public String getType() { return "Theropod"; }  
25 public String getDescription() { return "bipedal dinosaur usually with small forelimbs"; }  
26   
27 // override toString() method  
28 public String toString() {  
29 return("Type: " + this.getType() + " - " + this.getDescription() + "\n"  
30 + "Diet: " + super.getDiet() + "\n"  
31 + "Time period: " + super.getTime() + " million years ago\n"  
32 + "Location: " + super.getLocation() + "\n"  
33 + "Weight: " + super.getWeight() + "kg\n"  
34 + "Size: " + super.getSize() + "m\n"  
35 + "Walks on: " + super.getLegs() + " legs" + "\n"  
36 + "Eats: " + super.getFood() + "\n"  
37 + "Teeth: " + super.getTeeth() + "\n");  
38 }  
39 }

1 /\*\*  
 2 \* This is the CT\_Spinosaurus concrete class that extends C\_Theropod  
 3 \* (which extends Carnivore).  
 4 \*   
 5 \* This class:  
 6 \* ...has a constructor that sends the following variables to super   
 7 \* (C\_Theropod): time, location, size, legs, teeth, weight, food.  
 8 \* ...has appropriate getters that return name, meaning, and person.  
 9 \* ...overrides toString() to display all of the information about   
10 \* this dinosaur  
11 \* ...has printSpino() method that prints ASCII art of this dino.  
12 \*   
13 \* @author Stephanie Gremillion  
14 \* @version 17.0.2  
15 \* @since 2022/07/14  
16 \*/  
17   
18 public class CT\_Spinosaurus extends C\_Theropod {  
19 // constructor  
20 public CT\_Spinosaurus(String tim, String locate, double siz, int leg, String teet, int weigh, String foo) {  
21 super(tim, locate, siz, leg, teet, weigh, foo);  
22 }  
23   
24 // getters  
25 public String getName() { return "Spinosaurus"; }  
26 public String getMeaning() { return "thorn lizard"; }  
27 public String getPerson() { return "Stromer (1915)"; }  
28   
29 // override toString() method  
30 public String toString() {  
31 return("Name: " + this.getName() + ", which means \"" + this.getMeaning() + "\"\n"  
32 + "Named by: " + this.getPerson() + "\n"  
33 + "Type: " + super.getType() + " - " + super.getDescription() + "\n"  
34 + "Diet: " + super.getDiet() + "\n"  
35 + "Time period: " + super.getTime() + " million years ago\n"  
36 + "Location: " + super.getLocation() + "\n"  
37 + "Weight: " + super.getWeight() + "kg\n"  
38 + "Size: " + super.getSize() + "m\n"  
39 + "Walks on: " + super.getLegs() + " legs" + "\n"  
40 + "Eats: " + super.getFood() + "\n"  
41 + "Teeth: " + super.getTeeth() + "\n");  
42 }  
43   
44 // method to print spinosaurus picture  
45 public void printSpino() {  
46 System.out.println();  
47 System.out.println(" /###((%#%## ");  
48 System.out.println(" \*%(\*,,,,,,,\*,\*%%( #(##%#%%%%( ");  
49 System.out.println(" #%\*\*,,,,,\*,\*\*/\*\*(#&, #(/#(#(#(#(%%\*\* ");  
50 System.out.println(" &(\*\*(#%&&&&%###(#%%%&&( //((((/(###& ,%(#%%/,");  
51 System.out.println(" #%#%&(//(//#(%(###(#%#(##(#%((((/. \*(##,, (");  
52 System.out.println(" \*##%##/#((%%%%%%%%#(#%(###(#//#(//\* ,#(. ");  
53 System.out.println(" ##%%####((/####%#/(##((///((###/\*/((. , ");  
54 System.out.println(" ,(%%#%(#(#((((///((/\*\*/#(/###(((###(\*/((# ");  
55 System.out.println(" .##(####(((/\*///\*\*\*/\*/\*,//\*\*,/((/\*((////\*\*\*/\*\*#### ");  
56 System.out.println(" (%((///(//\*/\*\*\*\*//\*\*\*\*, .(/\*/\*\*\*\*/\*\*((((/ ## ");  
57 System.out.println(" /(((/// ,/(/((/\*(##( ");  
58 System.out.println(" (##(. \*\*/((\*//(##(\* ");  
59 System.out.println(" \*((, /((((, (###\* ");  
60 System.out.println(" (# (#%% %##% ");  
61 System.out.println("\*, (%%% ,####%##, ");  
62 System.out.println(" \*#(##((. ..,(%// ");  
63 System.out.println(" .,#&\*\*,\* ");  
64 System.out.println();  
65 }  
66 }

1 /\*\*  
 2 \* This is the CT\_Velociraptor concrete class that extends C\_Theropod  
 3 \* (which extends Carnivore).  
 4 \*   
 5 \* This class:  
 6 \* ...has a constructor that sends the following variables to super   
 7 \* (C\_Theropod): time, location, size, legs, teeth, weight, food.  
 8 \* ...has appropriate getters that return name, meaning, and person.  
 9 \* ...overrides toString() to display all of the information about   
10 \* this dinosaur  
11 \* ...has printVelo() method that prints ASCII art of this dino.  
12 \*   
13 \* @author Stephanie Gremillion  
14 \* @version 17.0.2  
15 \* @since 2022/07/14  
16 \*/  
17   
18 public class CT\_Velociraptor extends C\_Theropod {  
19 // constructor  
20 public CT\_Velociraptor(String tim, String locate, double siz, int leg, String teet, int weigh, String foo) {  
21 super(tim, locate, siz, leg, teet, weigh, foo);  
22 }  
23   
24 // getters  
25 public String getName() { return "Velociraptor"; }  
26 public String getMeaning() { return "quick plunderer"; }  
27 public String getPerson() { return "Osborn (1924)"; }  
28   
29 // override toString() method  
30 public String toString() {  
31 return("Name: " + this.getName() + ", which means \"" + this.getMeaning() + "\"\n"  
32 + "Named by: " + this.getPerson() + "\n"  
33 + "Type: " + super.getType() + " - " + super.getDescription() + "\n"  
34 + "Diet: " + super.getDiet() + "\n"  
35 + "Time period: " + super.getTime() + " million years ago\n"  
36 + "Location: " + super.getLocation() + "\n"  
37 + "Weight: " + super.getWeight() + "kg\n"  
38 + "Size: " + super.getSize() + "m\n"  
39 + "Walks on: " + super.getLegs() + " legs" + "\n"  
40 + "Eats: " + super.getFood() + "\n"  
41 + "Teeth: " + super.getTeeth() + "\n");  
42 }  
43   
44 // method to print velociraptor picture  
45 public void printVelo() {  
46 System.out.println();  
47 System.out.println(" .\*\*(%%% ");  
48 System.out.println(".\*\*\*\* #( ");  
49 System.out.println("./(\*/ . ");  
50 System.out.println(" ((\*// ");  
51 System.out.println(" .(//\*/ ");  
52 System.out.println(" /(\*\*/(( ");  
53 System.out.println(" ((//,/(# ///\* ");  
54 System.out.println(" ,#\*((\*/((\* .(((##\*\*\*\*(\* ");  
55 System.out.println(" #(\*(((\*\*((. \*(((#%((\*\*\*\*// ");  
56 System.out.println(" ##/\*/((,/((. ,((#(/#%%#%/\*/\*/ ");  
57 System.out.println(" (###,/(((///(/// (###\*(/(/((##(((\* ");  
58 System.out.println(" (##(#\*\*/((/(((//((/(((/(((/((#/(((((# ((((#((\*%#\*/#(###((/ ");  
59 System.out.println(" %#(#/(///((/(#(/((//((#(#((/(#(/(((/(&#(#(%##\*#,\*%.(##/(/# ");  
60 System.out.println(" ######((###(/(#(/(/(#(/#//##//(((/(((#(#(##( %,(( .. ");  
61 System.out.println(" /##(###%%#((#(/((###//#(##((/(/##(#(%#(###/,/,, (#(#&& ");  
62 System.out.println(" ###%##%##((##/#(%#(/(((/(((%%##%%%#(& #.%\*, \*%&&%@&%#%/. ");  
63 System.out.println(" &####(/(((/#%@%((((/(((%%%%#%#%%%&&&%/ /.(##%##, .%");  
64 System.out.println(" /##%##(/#/(%&###%//((%%%%%%%%# ,%&%#\*/ ");  
65 System.out.println(" #%#%##//(#%&&%%#(//##, ");  
66 System.out.println(" ,%((%#/(#&%&&&%#////. ");  
67 System.out.println(" %(#((((%&&&%&%%#////\* ");  
68 System.out.println(" #((/(##/(%%%%% #\*/// ");  
69 System.out.println(" /##(/####(%##% ./\*//\* ");  
70 System.out.println(" #(#((#%. \*#(( //\*\*(#/, ");  
71 System.out.println(" /#((, (((( (/@(/\*./%& ");  
72 System.out.println(" #//# ,###(##\* (/ .&%\* ");  
73 System.out.println(" ,(\*/, %%#####((&@ &% , ");  
74 System.out.println(" /(////##( %#%@% # @ ");  
75 System.out.println(" ((//((/#@&&& ");  
76 System.out.println(" \*(/@\* .& ");  
77 System.out.println();  
78 }  
79 }

1 /\*\*  
 2 \* This is the C\_Herrerasauridus concrete class that extend Carnivore.  
 3 \*   
 4 \* This class:  
 5 \* ...has a constructor that sends the following variables to super   
 6 \* (Carnivore): time, location, size, legs, teeth, weight, food.  
 7 \* ...has appropriate getters that return type "Herrerasauridus" as well as  
 8 \* returning the appropriate description.  
 9 \* ...overrides toString() to display the base information about this  
10 \* type of dino.  
11 \*   
12 \* @author Stephanie Gremillion  
13 \* @version 17.0.2  
14 \* @since 2022/07/14  
15 \*/  
16   
17 public class C\_Herrerasauridus extends Carnivore {  
18 // constructor  
19 public C\_Herrerasauridus(String tim, String locate, double siz, int leg, String teet, int weigh, String foo) {  
20 super(tim, locate, siz, leg, teet, weigh, foo);  
21 }  
22   
23 // getters  
24 public String getType() { return "Herrerasauridus"; }  
25 public String getDescription() { return "bipedal saurischin dinosaur, oldest known from the Late Triassic period"; }  
26   
27 // override toString() method  
28 public String toString() {  
29 return("Type: " + this.getType() + " - " + this.getDescription() + "\n"  
30 + "Diet: " + super.getDiet() + "\n"  
31 + "Time period: " + super.getTime() + " million years ago\n"  
32 + "Location: " + super.getLocation() + "\n"  
33 + "Weight: " + super.getWeight() + "kg\n"  
34 + "Size: " + super.getSize() + "m\n"  
35 + "Walks on: " + super.getLegs() + " legs" + "\n"  
36 + "Eats: " + super.getFood() + "\n"  
37 + "Teeth: " + super.getTeeth() + "\n");  
38 }  
39 }

1 /\*\*  
 2 \* This is the CH\_Herrerasaurus concrete class that extends C\_Herrerasauridus  
 3 \* (which extends Carnivore).  
 4 \*   
 5 \* This class:  
 6 \* ...has a constructor that sends the following variables to super   
 7 \* (C\_Herrerasauridus): time, location, size, legs, teeth, weight, food.  
 8 \* ...has appropriate getters that return name, meaning, and person.  
 9 \* ...overrides toString() to display all of the information about   
10 \* this dinosaur  
11 \* ...has printHerr() method that prints ASCII art of this dino.  
12 \*   
13 \* @author Stephanie Gremillion  
14 \* @version 17.0.2  
15 \* @since 2022/07/14  
16 \*/  
17   
18 public class CH\_Herrerasaurus extends C\_Herrerasauridus {  
19 // constructor  
20 public CH\_Herrerasaurus(String tim, String locate, double siz, int leg, String teet, int weigh, String foo) {  
21 super(tim, locate, siz, leg, teet, weigh, foo);  
22 }  
23   
24 // getters  
25 public String getName() { return "Herrerasaurus"; }  
26 public String getMeaning() { return "Herrera's lizard"; }  
27 public String getPerson() { return "Reig (1959)"; }  
28   
29 // override toString() method  
30 public String toString() {  
31 return("Name: " + this.getName() + ", which means \"" + this.getMeaning() + "\"\n"  
32 + "Named by: " + this.getPerson() + "\n"  
33 + "Type: " + super.getType() + " - " + super.getDescription() + "\n"  
34 + "Diet: " + super.getDiet() + "\n"  
35 + "Time period: " + super.getTime() + " million years ago\n"  
36 + "Location: " + super.getLocation() + "\n"  
37 + "Weight: " + super.getWeight() + "kg\n"  
38 + "Size: " + super.getSize() + "m\n"  
39 + "Walks on: " + super.getLegs() + " legs" + "\n"  
40 + "Eats: " + super.getFood() + "\n"  
41 + "Teeth: " + super.getTeeth() + "\n");  
42 }  
43   
44 // method to print herrerasaurus picture  
45 public void printHerr() {  
46 System.out.println();  
47 System.out.println(" @(#&&. , ( ");  
48 System.out.println(" %#(((/%&@%(## ");  
49 System.out.println(" @(/@&#(#&%#\*(/(/\*(/%(, ");  
50 System.out.println(" %&%@&(&(#(\*/#/ ,\*/(// \*\*((\*#(\*\*(,// ");  
51 System.out.println(" #%&&// .\*.&&&@@.&%#((%\*&\*&&&&#\*\*(&&%((##\*(/#(#/(( %##%#% @\*");  
52 System.out.println(" ##/% \* .. /&/\*&( (%(#\*(#(%%&&#/(%%((#((#/((#( &/##%%%& ");  
53 System.out.println(" \*(## #&#(/(((&%%#%\*&@/###/((%#(%##((%#\*(%&%%%# ");  
54 System.out.println(" %#( #(%@%&%@@%(#&&%&&@&%%#@@@@&##&&%%#####%, ");  
55 System.out.println(" \*,# /@\* &@@@@@%%#(%#%@&@&%@@@&&#((\*((## ");  
56 System.out.println(" #\* %# %#&&/,((@&&&#&(///( ");  
57 System.out.println(" %& &%% @@(%/.@@&%( ");  
58 System.out.println(" #&/ (#( &@@@#%( ");  
59 System.out.println(" @ %&@ @####%#/ ");  
60 System.out.println(" &% ##&%@@@%# ");  
61 System.out.println(" #@&% @&&%( ");  
62 System.out.println(" @&@@# @&% ");  
63 System.out.println(" @& %# (((# ");  
64 System.out.println(" %&\* /((## ");  
65 System.out.println();  
66 }  
67 }

1 /\*\*  
 2 \* This is the CH\_Staurikosaurus concrete class that extends C\_Herrerasauridus  
 3 \* (which extends Carnivore).  
 4 \*   
 5 \* This class:  
 6 \* ...has a constructor that sends the following variables to super   
 7 \* (C\_Herrerasauridus): time, location, size, legs, teeth, weight, food.  
 8 \* ...has appropriate getters that return name, meaning, and person.  
 9 \* ...overrides toString() to display all of the information about   
10 \* this dinosaur  
11 \* ...has printStaur() method that prints ASCII art of this dino.  
12 \*   
13 \* @author Stephanie Gremillion  
14 \* @version 17.0.2  
15 \* @since 2022/07/14  
16 \*/  
17   
18 public class CH\_Staurikosaurus extends C\_Herrerasauridus {  
19 // constructor  
20 public CH\_Staurikosaurus(String tim, String locate, double siz, int leg, String teet, int weigh, String foo) {  
21 super(tim, locate, siz, leg, teet, weigh, foo);  
22 }  
23   
24 // getters  
25 public String getName() { return "Staurikosaurus"; }  
26 public String getMeaning() { return "Southern Cross (after the star constellation visible from the Southern Hemisphere)"; }  
27 public String getPerson() { return "Colbert (1970)"; }  
28   
29 // override toString() method  
30 public String toString() {  
31 return("Name: " + this.getName() + ", which means \"" + this.getMeaning() + "\"\n"  
32 + "Named by: " + this.getPerson() + "\n"  
33 + "Type: " + super.getType() + " - " + super.getDescription() + "\n"  
34 + "Diet: " + super.getDiet() + "\n"  
35 + "Time period: " + super.getTime() + " million years ago\n"  
36 + "Location: " + super.getLocation() + "\n"  
37 + "Weight: " + super.getWeight() + "kg\n"  
38 + "Size: " + super.getSize() + "m\n"  
39 + "Walks on: " + super.getLegs() + " legs" + "\n"  
40 + "Eats: " + super.getFood() + "\n"  
41 + "Teeth: " + super.getTeeth() + "\n");  
42 }  
43   
44 // method to print staurikosaurus picture  
45 public void printStaur() {  
46 System.out.println();  
47 System.out.println(" (/ ");  
48 System.out.println(" /\*/\*/#\*(//((/((# ");  
49 System.out.println(" \*,\*(//\*\*\*/#,,/\*\*/(# ");  
50 System.out.println(" ./(( ,,,\*(#%%%(#(/((##((#####%%%\* ");  
51 System.out.println(" ,,\*/(\*/(/(///(////#((#/(((##(((% ");  
52 System.out.println(" /,,\*,\*\*////////(///(#(//(/(#/\*/\*\*((((#####%(. ");  
53 System.out.println(" \*/(###,(,/%/\*/\*\*\*\*\*\*\*//##(/\*//(%\*,,\*,,,,,\*\*\*\*,,\*\*\*\*\*\*\*/\*/((##");  
54 System.out.println(" /( /( \*/\*//\*\*\*/\*//((\*/\*//#/\*/(\*\* (");  
55 System.out.println(" \*(\*/ (/, /\*/((##\* ,/\*(#, ,#");  
56 System.out.println(" /(\* .(. /\*#( (/ ");  
57 System.out.println(" # /( . \*#\*(/ .(. ");  
58 System.out.println(" ./(((% \*//(. #( ");  
59 System.out.println(" ,/# # ");  
60 System.out.println(" .\*// ");  
61 System.out.println(" \*\*/( ");  
62 System.out.println(" \*/#( ");  
63 System.out.println();  
64 }  
65 }

1 /\*\*  
 2 \* This is the H\_Ornithiscian concrete class that extends Herbivore.  
 3 \*   
 4 \* This class:  
 5 \* ...has a constructor that sends the following variables to super   
 6 \* (Herbivore): time, location, size, legs, teeth, weight, food.  
 7 \* ...has appropriate getters that return type "Ornithiscian" as well as  
 8 \* returning the appropriate description.  
 9 \* ...overrides toString() to display the base information about this  
10 \* type of dino.  
11 \*   
12 \* @author Stephanie Gremillion  
13 \* @version 17.0.2  
14 \* @since 2022/07/14  
15 \*/  
16   
17 public class H\_Ornithiscian extends Herbivore {  
18 // constructor  
19 public H\_Ornithiscian(String tim, String locate, double siz, int leg, String teet, int weigh, String foo) {  
20 super(tim, locate, siz, leg, teet, weigh, foo);  
21 }  
22   
23 // getters  
24 public String getType() { return "Ornithiscian"; }  
25 public String getDescription() { return "herbivorous dinosaur with backward rotated pelvis"; }  
26   
27 // override toString() method  
28 public String toString() {  
29 return("Type: " + this.getType() + " - " + this.getDescription() + "\n"  
30 + "Diet: " + super.getDiet() + "\n"  
31 + "Time period: " + super.getTime() + " million years ago\n"  
32 + "Location: " + super.getLocation() + "\n"  
33 + "Weight: " + super.getWeight() + "kg\n"  
34 + "Size: " + super.getSize() + "m\n"  
35 + "Walks on: " + super.getLegs() + " legs" + "\n"  
36 + "Eats: " + super.getFood() + "\n"  
37 + "Teeth: " + super.getTeeth() + "\n");  
38 }  
39 }

1 /\*\*  
 2 \* This is the HO\_Agilisaurus concrete class that extends H\_Ornithiscian  
 3 \* (which extends Herbivore).  
 4 \*   
 5 \* This class:  
 6 \* ...has a constructor that sends the following variables to super   
 7 \* (H\_Ornithiscian): time, location, size, legs, teeth, weight, food.  
 8 \* ...has appropriate getters that return name, meaning, and person.  
 9 \* ...overrides toString() to display all of the information about   
10 \* this dinosaur  
11 \* ...has printAgil() method that prints ASCII art of this dino.  
12 \*   
13 \* @author Stephanie Gremillion  
14 \* @version 17.0.2  
15 \* @since 2022/07/14  
16 \*/  
17   
18 public class HO\_Agilisaurus extends H\_Ornithiscian {  
19 // constructor  
20 public HO\_Agilisaurus(String tim, String locate, double siz, int leg, String teet, int weigh, String foo) {  
21 super(tim, locate, siz, leg, teet, weigh, foo);  
22 }  
23   
24 // getters  
25 public String getName() { return "Agilisaurus"; }  
26 public String getMeaning() { return "agile lizard"; }  
27 public String getPerson() { return "Peng (1990)"; }  
28   
29 // override toString() method  
30 public String toString() {  
31 return("Name: " + this.getName() + ", which means \"" + this.getMeaning() + "\"\n"  
32 + "Named by: " + this.getPerson() + "\n"  
33 + "Type: " + super.getType() + " - " + super.getDescription() + "\n"  
34 + "Diet: " + super.getDiet() + "\n"  
35 + "Time period: " + super.getTime() + " million years ago\n"  
36 + "Location: " + super.getLocation() + "\n"  
37 + "Weight: " + super.getWeight() + "kg\n"  
38 + "Size: " + super.getSize() + "m\n"  
39 + "Walks on: " + super.getLegs() + " legs" + "\n"  
40 + "Eats: " + super.getFood() + "\n"  
41 + "Teeth: " + super.getTeeth() + "\n");  
42 }  
43   
44 // method to print agilisaurus picture  
45 public void printAgil() {  
46 System.out.println();  
47 System.out.println(" ,\*,, ");  
48 System.out.println(" .\*,,(. ");  
49 System.out.println(" ,\*,...\*/ ");  
50 System.out.println(" ..((/##,,(,,\*.\*,(\* ");  
51 System.out.println(" /%(#%%,\*(\*,,&,..\*. ..,,,,/(/ ");  
52 System.out.println(" ##/\*(,,(. .\*..,..\*,\*\*,,,,,\*/(. ");  
53 System.out.println(" .(#(. &/,,.,.,. \* ..../&((,#\*,,((%# ");  
54 System.out.println(" &#&(.,.. ., .,... ..#,\* ....#\*#\*,&/,(%@@@@ ");  
55 System.out.println(" .&.\*.\*/ ,\*\*\*..\*//#/,,,//,(,\*/&/\*,%,%(@#(/\*( ");  
56 System.out.println(" (/\*.\*\*\*. ,,.,/(( /\*/%(%# ");  
57 System.out.println(" \*(,,/ ,/\*\*%,,( \*//(/. ");  
58 System.out.println(" \*/#(/. (\*/. ,/\*@ ");  
59 System.out.println(" #\* .@ ,/ /\*,& ");  
60 System.out.println(" .# ,## ,((\* ");  
61 System.out.println(" @#/ &@. ");  
62 System.out.println();  
63 }  
64 }

1 /\*\*  
 2 \* This is the HO\_Lesothosaurus concrete class that extends H\_Ornithiscian  
 3 \* (which extends Herbivore).  
 4 \*   
 5 \* This class:  
 6 \* ...has a constructor that sends the following variables to super   
 7 \* (H\_Ornithiscian): time, location, size, legs, teeth, weight, food.  
 8 \* ...has appropriate getters that return name, meaning, and person.  
 9 \* ...overrides toString() to display all of the information about   
10 \* this dinosaur  
11 \* ...has printLes() method that prints ASCII art of this dino.  
12 \*   
13 \* @author Stephanie Gremillion  
14 \* @version 17.0.2  
15 \* @since 2022/07/14  
16 \*/  
17   
18 public class HO\_Lesothosaurus extends H\_Ornithiscian {  
19 // constructor  
20 public HO\_Lesothosaurus(String tim, String locate, double siz, int leg, String teet, int weigh, String foo) {  
21 super(tim, locate, siz, leg, teet, weigh, foo);  
22 }  
23   
24 // getters  
25 public String getName() { return "Lesothosaurus"; }  
26 public String getMeaning() { return "Lesotho lizard"; }  
27 public String getPerson() { return "Galton (1978)"; }  
28   
29 // override toString() method  
30 public String toString() {  
31 return("Name: " + this.getName() + ", which means \"" + this.getMeaning() + "\"\n"  
32 + "Named by: " + this.getPerson() + "\n"  
33 + "Type: " + super.getType() + " - " + super.getDescription() + "\n"  
34 + "Diet: " + super.getDiet() + "\n"  
35 + "Time period: " + super.getTime() + " million years ago\n"  
36 + "Location: " + super.getLocation() + "\n"  
37 + "Weight: " + super.getWeight() + "kg\n"  
38 + "Size: " + super.getSize() + "m\n"  
39 + "Walks on: " + super.getLegs() + " legs" + "\n"  
40 + "Eats: " + super.getFood() + "\n"  
41 + "Teeth: " + super.getTeeth() + "\n");  
42 }  
43   
44 // method to print lesothosaurus picture  
45 public void printLes() {  
46 System.out.println();  
47 System.out.println(" \*(/\*((\* ");  
48 System.out.println(" #((#,\*\*#\*/\*//( ");  
49 System.out.println(" (\*\*,\*\*\*///(# ");  
50 System.out.println(" ,,\*\*\*\*,\*\*\*//((. ");  
51 System.out.println(" \*,(\*\*/\*\*\*\*\*\*/\*///(///, ");  
52 System.out.println(" %,\*,,,,,,,,\*,\*,\*,\*/\*\*\*\*(( ");  
53 System.out.println(" &\*(, (\*,,,,,,,.,,,\*//,\*,,\*\*\*#//(#,. ,#( ");  
54 System.out.println(" \*( #( .#%\*,,,,,,,///\*\*\*,,\*\*,\*\*\*\*\*\*\*//(/((#(#/(\*(/((/////#\* ");  
55 System.out.println(" . (# ((\*#,,\*\*((#/,,/,,.,,,\*,//\*\*////((//, ");  
56 System.out.println(" #%/\*\*(, (((\*,...,/ ");  
57 System.out.println(" (\*,\* /#(/(/,,,,, ");  
58 System.out.println(" /\*,, (//\*((,# ");  
59 System.out.println(" ./,( ,((\* ");  
60 System.out.println(" %\*( %#(( ");  
61 System.out.println(" /, (&( ");  
62 System.out.println(" \*// ");  
63 System.out.println(" /(&#( ");  
64 System.out.println();  
65 }  
66 }

1 /\*\*  
 2 \* This is the H\_Sauropod concrete class that extends Herbivore.  
 3 \*   
 4 \* This class:  
 5 \* ...has a constructor that sends the following variables to super   
 6 \* (Herbivore): time, location, size, legs, teeth, weight, food.  
 7 \* ...has appropriate getters that return type "Sauropod" as well as  
 8 \* returning the appropriate description.  
 9 \* ...overrides toString() to display the base information about this  
10 \* type of dino.  
11 \*   
12 \* @author Stephanie Gremillion  
13 \* @version 17.0.2  
14 \* @since 2022/07/14  
15 \*/  
16   
17 public class H\_Sauropod extends Herbivore {  
18 // constructor  
19 public H\_Sauropod(String tim, String locate, double siz, int leg, String teet, int weigh, String foo) {  
20 super(tim, locate, siz, leg, teet, weigh, foo);  
21 }  
22   
23 // getters  
24 public String getType() { return "Sauropod"; }  
25 public String getDescription() { return "herbivorous dinosaur with long necks, tails, and legs"; }  
26   
27 // override toString() method  
28 public String toString() {  
29 return("Type: " + this.getType() + " - " + this.getDescription() + "\n"  
30 + "Diet: " + super.getDiet() + "\n"  
31 + "Time period: " + super.getTime() + " million years ago\n"  
32 + "Location: " + super.getLocation() + "\n"  
33 + "Weight: " + super.getWeight() + "kg\n"  
34 + "Size: " + super.getSize() + "m\n"  
35 + "Walks on: " + super.getLegs() + " legs" + "\n"  
36 + "Eats: " + super.getFood() + "\n"  
37 + "Teeth: " + super.getTeeth() + "\n");  
38 }  
39 }

1 /\*\*  
 2 \* This is the HO\_Brachiosaurus concrete class that extends H\_Sauropod  
 3 \* (which extends Herbivore).  
 4 \*   
 5 \* This class:  
 6 \* ...has a constructor that sends the following variables to super   
 7 \* (H\_Sauropod): time, location, size, legs, teeth, weight, food.  
 8 \* ...has appropriate getters that return name, meaning, and person.  
 9 \* ...overrides toString() to display all of the information about   
10 \* this dinosaur  
11 \* ...has printBrach() method that prints ASCII art of this dino.  
12 \*   
13 \* @author Stephanie Gremillion  
14 \* @version 17.0.2  
15 \* @since 2022/07/14  
16 \*/  
17   
18 public class HS\_Brachiosaurus extends H\_Sauropod {  
19 // constructor  
20 public HS\_Brachiosaurus(String tim, String locate, double siz, int leg, String teet, int weigh, String foo) {  
21 super(tim, locate, siz, leg, teet, weigh, foo);  
22 }  
23   
24 // getters  
25 public String getName() { return "Brachiosaurus"; }  
26 public String getMeaning() { return "arm lizard"; }  
27 public String getPerson() { return "Riggs (1903)"; }  
28   
29 // override toString() method  
30 public String toString() {  
31 return("Name: " + this.getName() + ", which means \"" + this.getMeaning() + "\"\n"  
32 + "Named by: " + this.getPerson() + "\n"  
33 + "Type: " + super.getType() + " - " + super.getDescription() + "\n"  
34 + "Diet: " + super.getDiet() + "\n"  
35 + "Time period: " + super.getTime() + " million years ago\n"  
36 + "Location: " + super.getLocation() + "\n"  
37 + "Weight: " + super.getWeight() + "kg\n"  
38 + "Size: " + super.getSize() + "m\n"  
39 + "Walks on: " + super.getLegs() + " legs" + "\n"  
40 + "Eats: " + super.getFood() + "\n"  
41 + "Teeth: " + super.getTeeth() + "\n");  
42 }  
43   
44 // method to print brachiosaurus picture  
45 public void printBrach() {  
46 System.out.println();  
47 System.out.println(" (/((##&&%. ");  
48 System.out.println(" (, &#\*/#%. ");  
49 System.out.println(" ((/###% ");  
50 System.out.println(" (#.&#%%, ");  
51 System.out.println(" \*/(\*(%%%. ");  
52 System.out.println(" //\*@##%& ");  
53 System.out.println(" /(/&###%% ");  
54 System.out.println(" ./(/((#%%( ");  
55 System.out.println(" (((#(((#%\* ");  
56 System.out.println(" ,/#/((((### ");  
57 System.out.println(" //#.#((#%%% ");  
58 System.out.println(" \*/(((((######. ");  
59 System.out.println(" (///(((/////(%, ");  
60 System.out.println(" \*////\*\*\*\*\*\*\*/(##& ");  
61 System.out.println(" \*\*\*,\*\*.,\*(\*//######( ");  
62 System.out.println(" ,,,.\*\*////(#(#(#####%& ");  
63 System.out.println(" \*,. ,\*/\*\*/((%%%###(######& ");  
64 System.out.println(" &#((/,.\*/#&#########%#(#%&&%%& ");  
65 System.out.println(" \*%&&.,/(@&%#####%%&&/(#%%&#%%###%\* ");  
66 System.out.println(" \*%&,/(@@&&&&%%@@@@%\*(%% (@&###%&%/, ");  
67 System.out.println(" &\*\*/(% %&@@,(%\* ");  
68 System.out.println(" ((\*(%. &&@/(# ");  
69 System.out.println(" ((\*(@ .&/(# ");  
70 System.out.println(" &/#% #\*\*(% ");  
71 System.out.println(" #@ ");  
72 System.out.println();  
73 }  
74 }

1 /\*\*  
 2 \* This is the HO\_Bellusaurus concrete class that extends H\_Sauropod  
 3 \* (which extends Herbivore).  
 4 \*   
 5 \* This class:  
 6 \* ...has a constructor that sends the following variables to super   
 7 \* (H\_Sauropod): time, location, size, legs, teeth, weight, food.  
 8 \* ...has appropriate getters that return name, meaning, and person.  
 9 \* ...overrides toString() to display all of the information about   
10 \* this dinosaur  
11 \* ...has printBell() method that prints ASCII art of this dino.  
12 \*   
13 \* @author Stephanie Gremillion  
14 \* @version 17.0.2  
15 \* @since 2022/07/14  
16 \*/  
17   
18 public class HS\_Bellusaurus extends H\_Sauropod {  
19 // constructor  
20 public HS\_Bellusaurus(String tim, String locate, double siz, int leg, String teet, int weigh, String foo) {  
21 super(tim, locate, siz, leg, teet, weigh, foo);  
22 }  
23   
24 // getters  
25 public String getName() { return "Bellusaurus"; }  
26 public String getMeaning() { return "fine lizard"; }  
27 public String getPerson() { return "Dong and Azuma (1990)"; }  
28   
29 // override toString() method  
30 public String toString() {  
31 return("Name: " + this.getName() + ", which means \"" + this.getMeaning() + "\"\n"  
32 + "Named by: " + this.getPerson() + "\n"  
33 + "Type: " + super.getType() + " - " + super.getDescription() + "\n"  
34 + "Diet: " + super.getDiet() + "\n"  
35 + "Time period: " + super.getTime() + " million years ago\n"  
36 + "Location: " + super.getLocation() + "\n"  
37 + "Weight: " + super.getWeight() + "kg\n"  
38 + "Size: " + super.getSize() + "m\n"  
39 + "Walks on: " + super.getLegs() + " legs" + "\n"  
40 + "Eats: " + super.getFood() + "\n"  
41 + "Teeth: " + super.getTeeth() + "\n");  
42 }  
43   
44 // method to print bellusaurus picture  
45 public void printBell() {  
46 System.out.println();  
47 System.out.println(" /#/((\*\* ");  
48 System.out.println(" ##\*/(#%#(//\* ");  
49 System.out.println(" (((//\* ");  
50 System.out.println(" /#(//\* ");  
51 System.out.println(" (#(/// .\*\*((/\*/(\* ");  
52 System.out.println(" (#(/((## \*\*//(\*\*///,\*\*\*,\*,\*/////(/,. ");  
53 System.out.println(" /##((\*///(((/(//\*\*////\*/\*///((#((\*/,,,, ((/,' '' '',(/\* ");  
54 System.out.println(" ###((\*((((##((///////#(\*(/((/(((((((//\*,,\* ,/ ");  
55 System.out.println(" ###(\*/(/(##%((/(((((#%//((//(#((#/##((/////\*\*,\*, \*,\*( ");  
56 System.out.println(" %%//(##%%##(#((#####(/(/((%%%#(\*/%%%#%##(((//((/////(#% ");  
57 System.out.println(" (/(#/ \*\*... ##(/(. ");  
58 System.out.println(" (#\*((# ((#((( ");  
59 System.out.println(" (/#//(% ,((%#(( ");  
60 System.out.println(" ((##/((% ,(((#(( ");  
61 System.out.println(" \*//(# #### ");  
62 System.out.println();  
63 }  
64 }