1. /\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*
2. AUTHOR: Usukhbayar Batbayar.
3. Mini-Project Suggestion 2: Dinosaur Pet Program, a program that simulates bio-engineered dinosaur pets that you must look after. by following all the instructions stated on each level.
4. \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/
5. import java.util.\*;
7. class projectmini
8. {
10. public static void main (String[] p) //Global strings to return to all methods
11. {
12. int thirst;
13. int hunger;
14. String name;
15. introduction();
16. name = askquestions();
18. thirst = Thirst(name);
19. hunger = Hunger();
20. answer(name);
21. angerDecision(thirst, hunger);
23. Pets s1 = new Pets(); // Records for pet.
24. int year = 2014;
25. s1 = setName (s1, name);
26. s1 = setRace (s1, "Pterosaurs");
27. s1 = setHunger (s1, hunger);
28. s1 = setThirst (s1, thirst);
29. s1 = setYear (s1, year);
31. print(" ");
32. print("Would you like to know Record information of your pet " + name +"? " +"Answer Yes/No");
33. Scanner scanner = new Scanner(System.in);
34. String reply;
35. reply = scanner.nextLine();
37. if (reply.equals("Yes".toLowerCase()))
38. {
39. print("Your pet is " + getRace(s1) + " and named " + getName(s1)+ ". His current hunger and thirst levels are " +getHunger(s1) +" and "+ getThirst(s1) +". He was born in " + getYear(s1) +".");
40. }
41. if (reply.equals("No"))
42. {
43. print("Record information is hidden.");
44. }
45. String[] emotions = new String[10];
46. emotions[0] = "Excessive Anger";
47. emotions[1] = "Depression";
48. emotions[2] = "Extreme Happiness";
49. rounds(thirst, hunger, emotions);
50. table(emotions, thirst, hunger);
52. game();
53. System.exit(0);
54. }
55. public static void game() // This has array of pets, loops and if statements.
56. {
57. Pets[] dinosaurs = {new Pets(), new Pets(), new Pets()}; // Array of new pets
58. dinosaurs[0] = setName(dinosaurs[0], "Rex");
59. dinosaurs[0] = setHunger(dinosaurs[0], 0);
60. dinosaurs[0] = setThirst(dinosaurs[0], 0);
61. dinosaurs[1] = setName(dinosaurs[1], "Tyra");
62. dinosaurs[1] = setHunger(dinosaurs[1], 0);
63. dinosaurs[1] = setThirst(dinosaurs[1], 0);
64. dinosaurs[2] = setName(dinosaurs[2], "Rob");
65. dinosaurs[2] = setThirst(dinosaurs[2], 0);
66. dinosaurs[2] = setHunger(dinosaurs[2], 0);
68. print(""); // Level 6 ADT with loops within loops
69. // pets();
70. boolean userwon = false; //boolean type to continue or end loop.
71. while(!userwon)
72. {
73. print("Intstruction: There are 2 rounds, 3 pets each has it's own hunger levels. You have to balance the food and water to win the game.");
74. print("Which pet would you want to start with?/Rex, Tyra, Rob/");
75. String chosenPet = getInput();
76. for (int i=1; i<3; i++) //rounds
77. {
78. print("Round "+i);
79. for(int j=0; j<3; j++) //nested for loops
80. {
81. print("What actions would you like to do to "+ getName(dinosaurs[j]) +"? (Feed/Water)");
82. String choice = getInput();
83. if(choice.toLowerCase().equals("feed"))
84. {
85. dinosaurs[j] =setHunger(dinosaurs[j], getHunger(dinosaurs[j]) + 1);
86. } else {
87. dinosaurs[j] =setThirst(dinosaurs[j], getThirst(dinosaurs[j]) + 1);
88. }
89. }
90. }
91. int hungerMult = getRandomInteger(1,5); //random int between 1 and 5
92. int thirstMult = getRandomInteger(1,5); //random int between 1 and 5
93. int[] score = new int[3];
94. for(int i=0; i<2; i++)
95. {
96. score[i] = getHunger(dinosaurs[i]) \* hungerMult + getThirst(dinosaurs[i]) \* thirstMult;
97. }
98. int maxscore = 0;
99. String winningdinoname = "";
100. for(int i=0; i<2; i++)
101. {
102. if(score[i]>maxscore)
103. {
104. maxscore = score[i];
105. winningdinoname=getName(dinosaurs[i]);
106. }
107. }
108. print("The winner is " +winningdinoname + " with the score of " + maxscore);
109. if(winningdinoname.toLowerCase().equals(chosenPet.toLowerCase()))
110. {
111. print("Congrats");
112. userwon = true;
113. }
114. else
115. {
116. print("You have lost. The game restarted");
117. print("");
118. }
120. }
121. }
122. public static String getInput()
123. {
124. Scanner scanner = new Scanner(System.in);
125. return scanner.nextLine();
126. }
127. public static void print (String print) // Print variable,we can directly use print to print output.
128. {
129. System.out.println(print);
130. }
131. public static void introduction() // The program starts here, first method.
132. {
133. print(" Hello!");
134. print(" Welcome to bio-engineered dinosaur pets.(âœ¿â— â€¿â— )");
135. }
137. public static String askquestions () // Stores user input under variable and prints it out throught the program.
138. {
139. print("What name would you like to give to your first pet?");
141. Scanner scanner = new Scanner(System.in);
142. String name;
143. name = scanner.nextLine();
145. print("Happy Birthday "+ name +" the Pterosaurs");
146. return name;
147. }
149. public static int Thirst(String name) // Variable Thirst stores values by input or calculation.
150. {
152. Random rand = new Random();
153. int thirstlevel = rand.nextInt(10) + 1;

156. print(name + "\'s thirst level is " +thirstlevel+ "/10");
158. return thirstlevel;
160. }
162. public static int getRandomInteger(int lowerBound, int upperBound)
163. {
164. Random rand = new Random();
165. return rand.nextInt(upperBound - lowerBound + 1) + lowerBound;
166. }
167. public static int Hunger() // Variable Hunger stores values by input or calculation.
168. {
169. Random rand = new Random();
170. int hungerlevel = rand.nextInt(10) + 1;
172. print(" "); // Level 4 - Bare pass methods
173. print("And her hunger level is " +hungerlevel+ "/10");
174. return hungerlevel;
175. }
177. public static void answer(String name) // Level 3, Now this method asks the user if they want to see anger level. Depending on the answer Yes or No, the next step is dependant.
178. {
179. Scanner scanner = new Scanner(System.in);
180. String ans;
181. print("Would you like to know " + name +"'s anger level?");
182. ans = scanner.nextLine();
184. System.out.println("Calculating...");
186. }
188. public static void angerDecision(int Thirst, int Hunger) // This will calculate anger level by Addition of Thirst and hunger level.
189. {
190. int angerlevel;
191. angerlevel = (Thirst+Hunger);
192. if (angerlevel <10) //The number here is compared that is stored from variables hungerleve and thristlevel
193. {
194. print("Her anger level is Serene..." + "Careful!");
195. }
196. else if (angerlevel <15)
197. {
198. print("Her anger level is Grouchy..." + "Call security!");
199. }
200. else // we expect this that is more or equal to than 15 or 16.
201. {
202. print("Her anger level is Dangerous..." + "GET OUTTA THERE NOW!");
204. }

207. }

210. //Level 3 F- Records and Getter Setter
211. // Getter methods for Pets record type
212. public static String getName (Pets s)
213. {
214. return s.name;
215. }
217. public static String getRace (Pets s)
218. {
219. return s.race;
220. }
222. public static int getHunger (Pets s)
223. {
224. return s.hunger;
225. }
226. public static int getThirst (Pets s)
227. {
228. return s.thirst;
229. }
231. public static int getYear (Pets s)
232. {
233. return s.year;
234. }
236. //Setter methods for Student record type
237. public static Pets setRace (Pets s, String racetype)
238. {
239. s.race = racetype;
240. return s;
241. }
243. public static Pets setName (Pets s, String dinoname)
244. {
245. s.name = dinoname;
246. return s;
247. }
248. public static Pets setHunger (Pets s, int hunger)
249. {
250. s.hunger = hunger;
251. return s;
252. }
253. public static Pets setThirst (Pets s, int thirst)
254. {
255. s.thirst = thirst;
256. return s;
257. }
259. public static Pets setYear (Pets s, int syear)
260. {
261. s.year = syear;
262. return s;
263. }
264. public static int rounds(int thirst, int hunger, String[] emotions) // Level 4 starts. We use Loops in this method and calculate them with other variables such as hunger and thirst to show how state of mind changes.
265. {
266. int round = 0;
268. for (int i=1; i<5; i++)
269. {
270. print(" ");
271. print("This is round " + i);
272. print("What actions would you like to do to your pet? (Feed/Sing/Water)");
274. impact(thirst, hunger, emotions);
275. }
276. return round;
277. }
278. public static void impact (int Thirst, int Hunger, String[] emotions) // The method rounds loops this 5 times, storing user input as Thirst, Hunger and State with Array names changing in the background. Will give you a table of answers.
279. {
280. Scanner scanner = new Scanner(System.in);
281. String ans = scanner.nextLine();
283. Random rand = new Random();
284. int Irritability = rand.nextInt(10) + 1;
286. int[] state = new int[10];
287. int i = 0;
289. if (ans.toLowerCase().equals("feed"))
290. {
291. int element = (Hunger)/2;
292. print("You have fed your pet. Hunger level decreased to " + element);
293. state[0] = element;
294. print("Emotional state changed from " + emotions[1] + " to " + emotions[2]);
295. }
296. else if (ans.toLowerCase().equals("sing"))
297. {
298. int element = (Irritability)/2;
299. print("You have sung to your pet. Irritability decreased to " + element);
300. state[1] = element;
301. print("Emotional state changed to " + emotions[i]);
302. }
303. else
304. {
305. int element = (Thirst)/2;
306. print("You have gave drink to your pet. Thirst level decreased to " + element);
307. state[2] = element;
308. print("Emotional state changed to " + emotions[i]);
309. }
311. }
313. public static void table(String[] emotions, int Thirst, int Hunger) //Makes a table from variables and array Thirst, Hunger and emotions.
314. {
315. for (int i=0; i<2; i++)
316. {
317. System.out.println(emotions[i] + "/" + Thirst + "/" + Hunger);
318. }
319. }
320. }
322. class Pets // represents a state of a pet
323. {
324. String name;
325. String race;
326. int hunger;
327. int thirst;
328. int year;
329. }