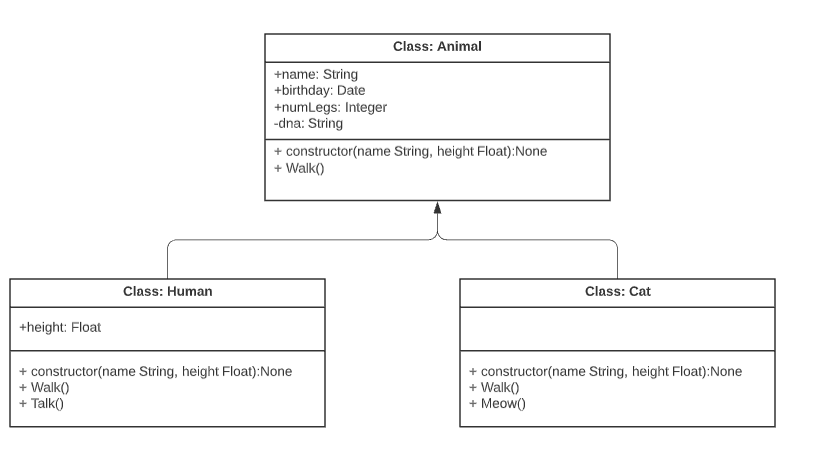
Student Name: Gabriel Solomon

Course: CIS123 Programming in Lava53

Date: 8/3/2120

# UML Class Diagram

Students should delete this UML example. Students should then copy/paste their own UML class diagram image into Microsoft Word. This is an image.



# Code

----------------------------App Class---------------------------

package com.assigment3;

import java.util.\*;

public class App

{

public static void main( String[] args )

{

Menu appMenu = new Menu();

Character command = '\_';

LinkedList<Panthera> catList = new LinkedList<>();

// loop until user quits

while (command != 'q') {

// print the menu

appMenu.print();

// get a command

System.out.print("Enter a command: ");

command = appMenu.getCommand();

// execute a commmand

appMenu.executeCommand(command, catList);

// move the cats to new positions in africa

appMenu.update(catList);

}

}

}

-------------------------Panthera Class-------------------------

package com.assigment3;

import java.util.Random;

/\*

\* Panthera base class that simulates GPS information

\*/

public class Panthera extends PantheraGPS{

/\*

TIP:

Students will need to add additional attributes and methods to complete this classes

implementation.

\*/

//add attributes

private int weight;

private Float speed;

// constructor

public Panthera(String name) {

// call the super-class (parent) to instatiate it

super(name);

// initialize attributes

this.setSpecies("panthera");

Random w = new Random();

this.weight = w.nextInt(100-10) + 10;

//setSpeed();

}

public void setSpeed(){

Random s = new Random();

this.speed = minSpeed + s.nextFloat() \* (maxSpeed - minSpeed);

}

public int getWeight(){

return this.weight;

}

public Float getSpeed(){

return this.speed;

}

// serializes attributes into a string

@Override // override superclass method

public String toString() {

String s;

// since the object is complex, we return a JSON formatted string

s = "{ ";

s += "name: " + this.name();

s += ", ";

s += "species: " + this.species();

s += ", ";

s += "longitude: " + this.longitude();

s += ", ";

s += "latitude: " + this.latitude();

s += ", ";

s += "weight: " + this.getWeight() + " lbs";

s += ", ";

s += "speed: " + this.getSpeed();

s += " }";

return s;

}

public void roar(){

System.out.println("Rrrrrrrrrrroooooooaaaaarrrrr!");

}

}

-------------------------PantheraGPS Class----------------------

package com.assigment3;

import java.util.Random;

/\*

\* Panthera base class that simulates GPS information

\*/

public class PantheraGPS {

// constants

private final Integer maxLongitude = 1000;

private final Integer maxLatitude = 1000;

protected final float minSpeed = 0f;

protected final float maxSpeed = 50.0f;

// attributes

private String name;

private String species;

private Float longitude;

private Float latitude;

private Random longitudeRandom;

private Random latitudeRandom;

// constructor

public PantheraGPS(String name) {

// initialize attributes

this.name = name;

this.species = "pantheraGPS";

// seed the random number generators for repeatable results

this.longitudeRandom = new Random();

this.longitudeRandom.setSeed(this.seed(name + "longitude"));

this.latitudeRandom = new Random();

this.latitudeRandom.setSeed(this.seed(name + "latitude"));

// move the panthera into it's initial position

this.longitude = longitudeRandom.nextFloat() \* maxLongitude;

this.latitude = latitudeRandom.nextFloat() \* maxLatitude;

}

// serializes attributes into a string

public String toString() {

String s;

// since the object is complex, we return a JSON formatted string

s = "{ ";

s += "name: " + name;

s += ", ";

s += "species: " + this.species();

s += ", ";

s += "longitude: " + this.longitude();

s += ", ";

s += "latitude: " + this.latitude();

s += " }";

return s;

}

// getters

public String name() {

return this.name;

}

public String species() {

return this.species;

}

public void setSpecies(String species) {

this.species = species;

}

// make a seed, based on the name

private Integer seed(String s) {

Integer seed = 0;

for (Integer i = 0; i < s.length() ; i++) {

char ch = s.charAt(i);

seed += (int) ch;

}

return seed;

}

public void move() {

this.longitude += longitudeRandom.nextFloat() \* maxSpeed;

this.latitude += latitudeRandom.nextFloat() \* maxSpeed;

}

// longitude of the panthera

public Float longitude() {

return longitude;

}

// latitude of the panthera

public Float latitude() {

return latitude;

}

}

-------------------------Menu Class---------------------------

package com.assigment3;

/\*

\* Menu class for the african big cat app

\*/

import java.util.LinkedList;

import java.util.Scanner;

public class Menu {

// attributes

private Scanner input;

// constructor

public Menu() {

// initialize attributes

this.input = new Scanner(System.in);

}

// prints the menu

public void print() {

printLine();

System.out.println("African Big Cats App");

printLine();

printCommand('c',"[C]reates a big cat");

printCommand('d',"[D]eletes a big cat");

printCommand('f',"[F]inds a big cat");

printCommand('l',"[L]ists all big Cats");

printCommand('r',"[R]isk report");

printCommand('w',"[W]arning report");

printCommand('q',"[Q]uits");

printLine();

}

private static void printLine() {

System.out.println("----------------------------------------------------------" );

}

private static void printCommand(Character command, String desc) {

System.out.printf("%s\t%s\n", command, desc);

}

// get first character from input

public Character getCommand() {

Character command = '\_';

input = new Scanner(System.in);

String rawInput = input.nextLine();

if (rawInput.length() > 0) {

rawInput = rawInput.toLowerCase();

command = rawInput.charAt(0);

}

return command;

}

// command switch

public Boolean executeCommand(Character command, LinkedList<Panthera> catList) {

Boolean success = true;

/\*

TIP:

In this area of the code, the additional commands need to be created and added.

\*/

switch(command) {

case 'c':

executeCreate(catList);

break;

/\* case 'd':

executeDelete(catList);

break;

case 'f':

executeFind(catList);

break;

\*/

case 'l':

executeList(catList);

break;

case 'q':

executeQuit();

break;

default:

System.out.println("ERROR: Unknown commmand");

success = false;

}

return success;

}

// update the position of all the cats

public void update(LinkedList<Panthera> catList) {

// update by moving all the cats

for (Panthera cat: catList) {

cat.move();

}

}

// quit the app

public void executeQuit() {

// close the scannner

input.close();

System.out.println();

printLine();

System.out.println("Thank you for using the African Big Cats App!");

printLine();

System.out.println();

}

public Panthera getNewCat(String name) {

/\*

TIP:

In this area of the code, students need to get input from the user for the type of cat

and create the correct type.

Currently, the code always create a Tiger. But, support for Lions and Jaguars

also needs to be added.

\*/

Panthera result = null;

while (result == null) {

System.out.println("Enter a 1 for Tiger, 2 for Lion, 3 for Jaguard");

Integer userChoose = input.nextInt();

switch (userChoose){

case 1:

result = new Tiger(name);

break;

case 2:

result = new Lions(name);

break;

case 3:

result = new Jaguards(name);

break;

default:

System.out.println("please 1, 2 or 3 only!");

System.out.println();

break;

}

}

return result;

}

// create a cat, if it's unique

public void executeCreate(LinkedList<Panthera> catList) {

/\*

TIP:

In this area of the code, students would need to add in checking if thecat name

already exists in order to prevent duplicates

\*/

// get the name

System.out.print("Enter a name for the big cat to create: ");

String name = input.nextLine();

System.out.println();

for (int i = 0; i < catList.size();){

if (catList.get(i).name().equals(name)){

System.out.println("This name is existed. Please choose another one!");

System.out.println();

System.out.print("Enter a name for the big cat to create: ");

name = input.nextLine();

System.out.println();

i = 0;

} else {

i++;

}

}

Panthera cat = getNewCat(name);

catList.add(cat);

System.out.println();

System.out.println("STATUS: " + name + " has been added.");

}

// list all big cats

public void executeList(LinkedList<Panthera> catList) {

System.out.println();

printLine();

System.out.println("African Big Cat List");

printLine();

Panthera cat;

if (catList.size() > 0) {

for (Integer i = 0; i < catList.size(); i++) {

cat = catList.get(i);

System.out.println(cat);

}

} else {

System.out.println("There are no African Big Cats. :(");

}

System.out.println();

}

/\*

TIP:

Additional methods and functionality need to be added to this class.

\*/

}

-------------------------Lion Class---------------------------

package com.assigment3;

public class Lions extends Panthera{

// constructor

public Lions(String name) {

// call the super-class (parent) to instatiate it

super(name);

// initialize attributes

this.setSpecies("lions");

}

// serializes attributes into a string

@Override // override superclass method

public String toString() {

String s;

// since the object is complex, we return a JSON formatted string

s = "{ ";

s += "name: " + name();

s += ", ";

s += "species: " + species();

s += ", ";

s += "longitude: " + longitude();

s += ", ";

s += "latitude: " + latitude();

s += ", ";

s += "weight: " + getWeight() + " lbs";

s += ", ";

s += "speed: " + getSpeed();

s += ", ";

s += "fur: " + fur();

s += " }";

return s;

}

public String fur() {

return "mane";

}

}

-------------------------Tiger Class---------------------------

package com.assigment3;

/\*

\* Panthera base class that simulates GPS information

\*/

public class Tiger extends Panthera{

// constructor

public Tiger(String name) {

// call the super-class (parent) to instatiate it

super(name);

// initialize attributes

this.setSpecies("tiger");

}

// serializes attributes into a string

@Override // override superclass method

public String toString() {

String s;

// since the object is complex, we return a JSON formatted string

s = "{ ";

s += "name: " + name();

s += ", ";

s += "species: " + species();

s += ", ";

s += "longitude: " + longitude();

s += ", ";

s += "latitude: " + latitude();

s += ", ";

s += "weight: " + getWeight() + " lbs";

s += ", ";

s += "speed: " + getSpeed();

s += ", ";

s += "fur: " + fur();

s += " }";

return s;

}

public String fur() {

return "stripes";

}

}

-------------------------Jaguards Class-------------------------

package com.assigment3;

public class Jaguards extends Panthera {

// constructor

public Jaguards(String name) {

// call the super-class (parent) to instatiate it

super(name);

// initialize attributes

this.setSpecies("jaguards");

}

// serializes attributes into a string

@Override // override superclass method

public String toString() {

String s;

// since the object is complex, we return a JSON formatted string

s = "{ ";

s += "name: " + name();

s += ", ";

s += "species: " + species();

s += ", ";

s += "longitude: " + longitude();

s += ", ";

s += "latitude: " + latitude();

s += ", ";

s += "weight: " + getWeight() + " lbs";

s += ", ";

s += "speed: " + getSpeed();

s += ", ";

s += "fur: " + fur();

s += ", ";

s += "sleepsInTrees: " + sleepsInTrees();

s += " }";

return s;

}

public String fur() {

return "spots";

}

public boolean sleepsInTrees(){

return true;

}

}

# Demo Video of software running

* Students should delete this youtube link. Students should then copy/paste the unlisted and accessible youtube demo video of their software running here.
* Unlisted YouTube Video demo of your software running.
* Test your video for accessibility (that I can access it) prior to submission by asking someone you know to click on the link to test it or by using Chrome’s Incognito feature.

<https://youtube.com/watch?v=2uCXIbkbDSEkjk23k3>

# Image of GitHub Desktop History of Commits

Text

Description automatically generated