CS115 Lab 10 Key

1. (2 points) Break a problem down into classes with attributes and methods.

Weather class Attributes: low temp, high temp, sky condition Methods: accessors and Mutators as usual, with data verification on range of temps, and on sky condition string

Date Forecasts Class Attributes: date that the forecast is for collection of 3 "Weather" objects (3 days prior, 2 days prior, prior), and one "Weather" object for the actual weather. Methods: accessors and mutators for the three "Weather" objects and actual "Weather" object, and a method to output accuracy of predictions results for this "Date Forecasts" object

Month class Attributes a collection of 30 days worth of "Date Forecasts" objects counter for number of objects currently in the collection Methods: methods to add, find, delete, replace a WeatherForecast object in the collection methods to calculate and output statistics on the collection

2. (3 points) Design, code and test programs that require arrays.

NO CHANGES NEEDED TO BobsLife

public class BobsLife {

private String location;

private int hunger;

private int fitness;

private int dollars;

private int time;

private boolean isDead;

private boolean inJail;

private final String DEFAULT\_LOCATION="home";

public BobsLife (String l, int h, int f, int d) {

time = 0;

isDead = false;

inJail = false;

setLocation(l);

hunger=h;

fitness=f;

dollars=d;

}

private void setLocation(String l) {

if(!isDead && !inJail) {

if (l!=null && (l.equals("home") || l.equals("work") || l.equals("gym")))

location = l;

else location=DEFAULT\_LOCATION;

}

}

public void move(String to) {

setLocation(to);

}

public void nextTime() {

time++;

if (inJail || isDead) return;

if (location.equals("home")) atHome();

else if (location.equals("gym")) atGym();

else if (location.equals("work")) atWork();

}

private void atHome() {

hunger = Math.max(0, hunger - 3);

dollars -= 1;

checkState();

}

private void atWork() {

hunger+= 2;

dollars += 3;

fitness -= 1;

checkState();

}

private void atGym() {

hunger += 3;

dollars -= 2;

fitness += 2;

checkState();

}

private void checkState() {

if (dollars < 0) inJail = true;

if (hunger > 6) isDead = true;

if (fitness == 0) isDead = true;

}

public String toString() {

return "Time: "+time+" - location:"+location+", hunger:"+hunger+", dollars:"+dollars+"a fitness:"+fitness+" ("+getStatus()+")";

}

public String getStatus() {

if (isDead)

return ("deceased");

if (inJail)

return ("in jail");

return "alive and well";

}

}

CHANGES NEEDED IN LongLiveBob

public class LongLiveBob {

public static void main(String[] args) {

int stratPos = 0;

final String[] STRATEGY = new String[] {

"home",

"home",

"home",

"work",

"work",

"work",

"home",

"home",

"gym",

"gym",

"home",

"work"

};

BobsLife b = new BobsLife("home", 0, 4, 5);

System.out.println(b);

for(int i = 0; i < 1000; i++) {

b.move(STRATEGY[stratPos]);

b.nextTime();

System.out.println(b);

stratPos++;

if (stratPos >= STRATEGY.length) stratPos = 0;

}

}

}

3. (5 points) Write a program that utilizes reading from file with a Loop (hasNext to recognize EOF). Design, code and test programs that require arrays.

public class StudentAnswerSheet {

private String name;

private char [] answers;

private static final char BLANK='?';

private static final double CORRECT\_POINTS=1., BLANK\_POINTS=0., WRONG\_POINTS=-.25;

private static char [] key;

public static void setKey(char [] answerKey) {

if (answerKey!=null) {

key=new char[answerKey.length];

for (int i=0;i<answerKey.length;i++) key[i]=answerKey[i];

}

else key=new char[0];

}

public StudentAnswerSheet (String n, char [] a) {

if (n!=null && a!=null) {

answers=new char[a.length];

name=n;

for (int i=0;i<a.length;i++) answers[i]=a[i];

}

else {

name="";

answers=new char[0];

}

}

public String getName () { return name; }

public double getScore () {

double total=0;

if (key.length == answers.length) {

for (int i=0;i<key.length;i++) {

if (answers[i]==BLANK) total+=BLANK\_POINTS;

else if (answers[i]==key[i]) total+=CORRECT\_POINTS;

else total+=WRONG\_POINTS;

}

}

else total=Double.NEGATIVE\_INFINITY;

return total;

}

public String toString () {

String temp = name;

for (int i=0; i<answers.length;i++) temp = temp + ' ' + answers[i];

temp=temp+" score="+getScore();

return temp;

}

}

import java.util.\*;

import java.io.\*;

public class GradeStudents {

public static void main(String[] args) throws IOException {

File inputFile = new File("answers.txt");

Scanner in = new Scanner(inputFile);

boolean keyDone=false;

double highScore=0; String highStudent="";

while (in.hasNext()) {

String [] data=in.nextLine().split(",");

String name=data[0];

char [] ans = new char[data.length - 1];

for (int i=1; i<data.length; i++) ans[i-1] = data[i].charAt(0);

if (!keyDone) {

StudentAnswerSheet.setKey(ans);

keyDone=true;

}

else {

StudentAnswerSheet a = new StudentAnswerSheet(name, ans);

double s = a.getScore();

System.out.println(a);

if (s>highScore) {

highScore=s;

highStudent=name;

}

else if (s==highScore) highStudent=highStudent+" "+name;

}

}

System.out.println("Best Student(s): "+highStudent);

}

}

4. (5 points) Write a class that requies class attributes/methods and iteration.

import java.text.DecimalFormat;

public class Particle {

//instance variables

private double a, b, c, d;

private static final DecimalFormat THREE\_FORMAT = new DecimalFormat("##0.000");

private static double maxVelocity=Double.NEGATIVE\_INFINITY;

//non-default constructor

public Particle(double newA, double newB, double newC, double newD) {

setA(newA);

setB(newB);

setC(newC);

setD(newD);

}

//accessor methods

public double getA() { return a; }

public double getB() { return b; }

public double getC() { return c; }

public double getD() { return d; }

//mutator methods

public void setA(double newA) { a=newA; }

public void setB(double newB) { b=newB; }

public void setC(double newC) { c=newC; }

public void setD(double newD) { d=newD; }

public void table(double start, double end, double increment) {

double i=start;

double location, velocity, prevLocation=0;

System.out.println("Time\tLoc\tVel");

while ( i <= end ) {

location = a\*Math.pow(i, 4) + b\*Math.pow(i, 3) + c\*Math.pow(i, 2) + d\*i;

velocity = 4\*a\*Math.pow(i, 3) + 3\*b\*Math.pow(i, 2) + 2\*c\*i;

System.out.print(i + "\t" + THREE\_FORMAT.format(location) + "\t"+ THREE\_FORMAT.format(velocity));

if (i>start && Math.abs(prevLocation-location)>1)

System.out.print(" More than 1 unit movement");

prevLocation = location;

if (velocity>maxVelocity) maxVelocity=velocity;

System.out.println();

i = i + increment;

}

}

public static double getMaxVelocity() { return maxVelocity; }

public String toString() {

return "location(t) = " +a+"t^4+"+b+"t^3+"+c+"t^2+"+d+"t";

}

}

5. (5 points) Write a class that requies class attributes/methods and iteration.

public class Vector {

public static void main(String[] args) {

int [] v1 = { 1, 3, 5};

int [] v2 = { -1, -3, -5};

int [] v3 = { 1, 2, 3, 4};

int [] v4 = { 1, 1, 1, 1};

System.out.println(vectorToString(v1)+"+"+vectorToString(v2)+"="+vectorToString(add(v1, v2)));

System.out.println(vectorToString(v1)+"+"+vectorToString(v3)+"="+vectorToString(add(v1, v3)));

System.out.println(vectorToString(v2)+"+"+vectorToString(v3)+"="+vectorToString(add(v2, v3)));

System.out.println(vectorToString(v3)+"+"+vectorToString(v4)+"="+vectorToString(add(v3, v4)));

System.out.println(vectorToString(v1)+"."+vectorToString(v2)+"="+dotProduct(v1, v2));

System.out.println(vectorToString(v1)+"."+vectorToString(v3)+"="+dotProduct(v1, v3));

System.out.println(vectorToString(v2)+"."+vectorToString(v3)+"="+dotProduct(v2, v3));

System.out.println(vectorToString(v3)+"."+vectorToString(v4)+"="+dotProduct(v3, v4));

}

public static int [] add(int [] a, int [] b) {

int [] total=null;

if (a.length == b.length){

total = new int[a.length];

for (int i=0; i<a.length; i++)

total[i]=a[i]+b[i];

}

else System.err.print("Sum not defined, vectors different length");

return total;

}

public static int dotProduct(int [] a, int [] b) {

int total=0;

if (a.length == b.length)

for (int i=0; i<a.length; i++)

total=total+a[i]\*b[i];

else System.err.print("Dot Product not defined, vectors different length");

return total;

}

public static String vectorToString(int [] a) {

String x="<";

if (a!=null) {

x=x+a[0];

for (int i=1; i<a.length; i++) x=x+","+a[i];

}

x=x+">";

return x;

}

}