Weslyn Wagner EE 5343 UTSA: Java Assignment #3 10/08/2016

|  |  |
| --- | --- |
| Class Complex | |
| Member variables | - double real  - double imag |
| Constructors | +public Complex()  +public Complex(double real)  +public Complex(double real, double imag) |
| Setters/Getters | +set\_real(double real); +get\_real()  +set\_imag(double imag); +get\_imag() |
| Member methods | +display() +add(Complex p)  +magnitude() +subtract(Complex p)  +isReal() +multiply(Complex p)  +isImaginary() +divide(Complex p)  +conjugate() +equals(Complex p) |

/\* Weslyn Wagner - zfs119

\* Sept8

\*

\* Version 1.0 2016/10/08

\*

\* EE 5343 UTSA

\*/

import java.lang.Math.\*;

class ComplexRun{

public static void main(String[] args){

//instantiate some new Complex members

Complex one = new Complex();

Complex two = new Complex(1.0);

Complex three = new Complex(1.0,-2.0);

Complex four = new Complex(1.0,2.0);

Complex five = new Complex(3.0,4.0);

Complex six = new Complex(0.0,4.0);

//test out the Complex class member methods

one.display();

System.out.println("this is mag: "+ three.magnitude());

System.out.println("Is two real? "+ two.isReal());

System.out.println("Is four real? "+ four.isReal());

System.out.println("Is two imag? "+ two.isImaginary());

System.out.println("Is six imag? "+ six.isImaginary());

System.out.println("five's conjugate: ");

five.conjugate();

System.out.println("three's conjugate: ");

three.conjugate();

System.out.println("does one eqaul two? "+ one.equals(two));

System.out.println("does three equal four? "+ three.equals(four));

System.out.println("add ");

three.add(five);

System.out.println("subtract ");

three.subtract(five);

System.out.println("multiply ");

three.multiply(five);

System.out.println("divide ");

three.divide(five);

}

}

class Complex{

//member variables

private double real;

private double imag;

//constructors

public Complex(){

real = 0.0;

imag = 0.0;

}

public Complex(double real){

this.real = real;

imag = 0.0;

}

public Complex(double real, double imag){

this.real = real;

this.imag = imag;

}

//getters & setters

public void set\_real(double real){

this.real = real;

}

public double get\_real(){

return real;

}

public void set\_imag(double imag){

this.imag = imag;

}

public double get\_imag(){

return imag;

}

//member methods

private double i;

private double j;

public void display(){

System.out.println(real+" + "+imag+"j");

}

public double magnitude(){

double i = this.real;

double j = this.imag;

return Math.sqrt((i\*i)+(j\*j));

}

public boolean isReal(){

if(this.real != 0 && this.imag ==0){

return true;}

else{return false;}

}

public boolean isImaginary(){

if(this.real == 0 && this.imag !=0){

return true;}

else{return false;}

}

public void conjugate(){

if(-1\*imag<0){

System.out.println(real+" - "+Math.abs((-1\*imag))+"j");

}

else{

System.out.println(real+" + "+Math.abs((-1\*imag))+"j");

}

}

public boolean equals(Complex p){

boolean test;

if(this.real == p.real && this.imag == p.imag){

test = true;

}

else{test = false;}

return test;

}

public void add(Complex p){

i = this.real + p.real;

j = this.imag + p.imag;

System.out.println(i+" + "+j+"j");

}

public void subtract(Complex p){

i = this.real - p.real;

j = this.imag - p.imag;

System.out.println(i+" + "+j+"j");

}

public void multiply(Complex p){

i = this.real\*p.real - this.imag\*p.imag;

j = this.imag\*p.real + this.real\*p.imag;

System.out.println(i+" + "+j+"j");

}

public void divide(Complex p){

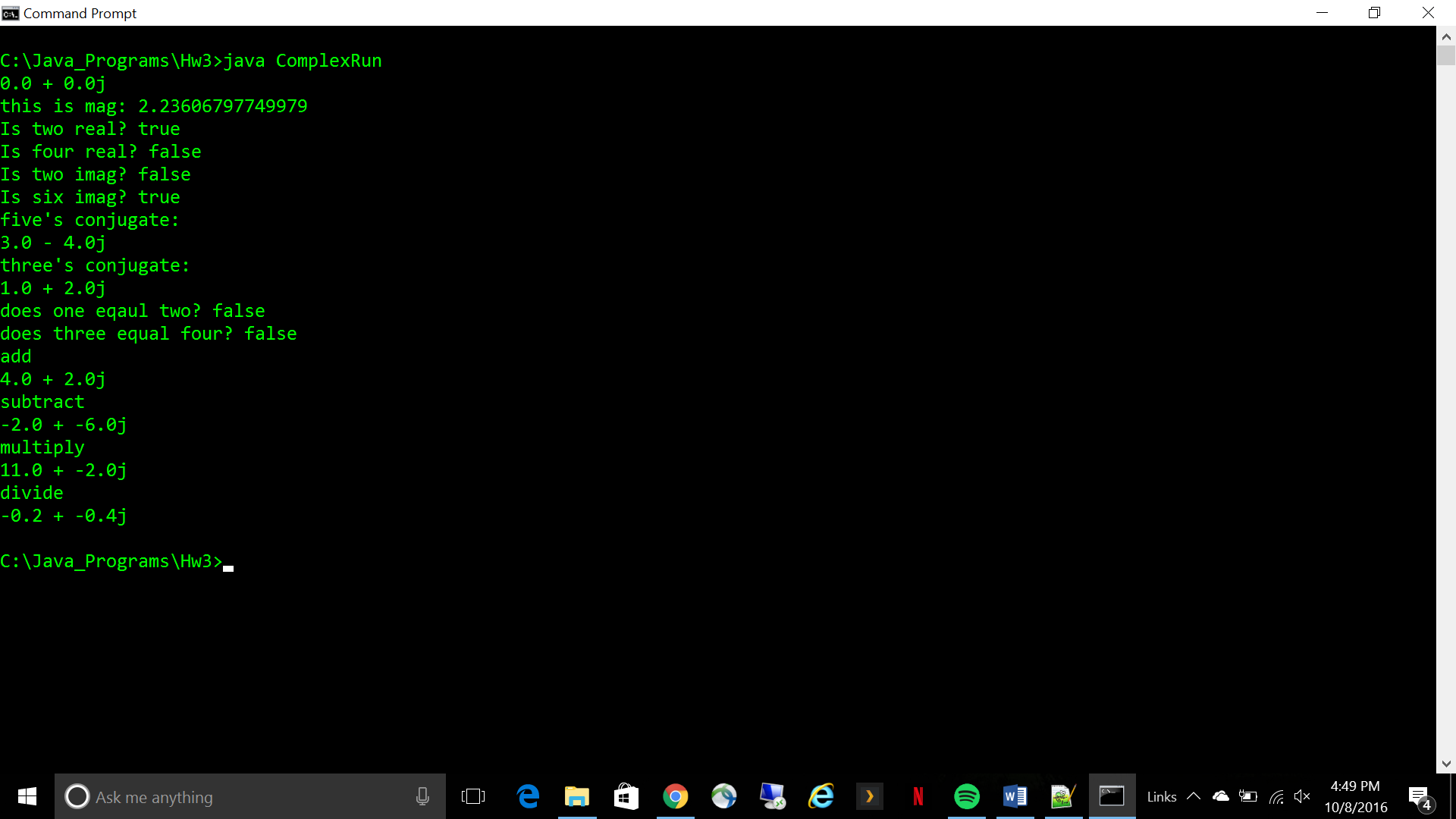
i= (this.real\*p.real + this.imag\*p.imag)/((p.real\*p.real)+(p.imag\*p.imag));

j= (this.imag\*p.real - this.real\*p.imag)/((p.real\*p.real)+(p.imag\*p.imag));

System.out.println(i+" + "+j+"j");

}

}



|  |  |
| --- | --- |
| Class Planet | |
| Member variables | - String name - double size - String color - boolean isVisible - int moons - double lightyearsFromSun |
| Constructors | Planet()  Planet(String name)  Planet(String name, double size)  Planet(String name, double size, String color)  Planet(String name, double size, String color, boolean isVisible)  Planet(String name, double size, String color, boolean isVisible, int moons)  Planet(String name, double size, String color, boolean isVisible, int moons, double lightyearsFromSun) |
| Setters/Getters | +set\_name(String name); +get\_name()  +set\_size(double size); +get\_size()  +set\_color(String color); +get\_color()  +set\_isVisible(Boolean isVisible); +get\_isVisible()  +set\_moons(int moons), +get\_moons()  +set\_lightyearsFromSun(double lightyearsFromSun); +get\_lightyearsFromSun() |
| Member methods | +explode()  +closerToSun(double distance)  +changeColor(String color)  +addMoons(int moons)  +hide() |

/\* Weslyn Wagner - zfs119

\* Sept8

\*

\* Version 1.0 2016/10/08

\*

\* EE 5343 UTSA

\*/

import java.lang.\*;

class Planets {

public static void main(String[] args) {

//instantiate a new planet called earth

Planet earth = new Planet("Earth",100,"green",true,1,5.9);

//move earth a little closer to the sun

earth.closerToSun(3);

//move earth too close to the sun

earth.closerToSun(4);

//change its color to blue

earth.changeColor("blue");

//add some moons to the one that's already there

earth.addMoons(2);

//check and see if earth is visible

System.out.println("Is earth visible?");

System.out.println(earth.get\_isVisible());

//hide earth and check and see if it's visible

earth.hide();

System.out.println("Is earth visible?");

System.out.println(earth.get\_isVisible());

//explode the earth

earth.explode();

}

}

class Planet{

//member variables

private String name;

private double size;

private String color;

private boolean isVisible;

private int moons;

private double lightyearsFromSun;

//constructors

public Planet(){

name = "";

size = 0.0;

color = "nuetral";

isVisible = true;

moons = 0;

lightyearsFromSun = 0.0;

}

public Planet(String name){

this.name =name;

size = 0.0;

color = "nuetral";

isVisible = true;

moons = 0;

lightyearsFromSun = 0.0;

}

public Planet(String name, double size){

this.name = name;

this.size = size;

color = "nuetral";

isVisible = true;

moons = 0;

lightyearsFromSun = 0.0;

}

public Planet(String name, double size, String color){

this.name = name;

this.size = size;

this.color = color;

isVisible = true;

moons = 0;

lightyearsFromSun = 0.0;

}

public Planet(String name, double size, String color, boolean isVisible){

this.name = name;

this.size = size;

this.color = color;

this.isVisible = isVisible;

moons = 0;

lightyearsFromSun = 0.0;

}

public Planet(String name, double size, String color, boolean isVisible, int moons){

this.name = name;

this.size = size;

this.color = color;

this.isVisible = isVisible;

this.moons = moons;

lightyearsFromSun = 0.0;

}

public Planet(String name, double size, String color, boolean isVisible, int moons, double lightyearsFromSun){

this.name = name;

this.size = size;

this.color = color;

this.isVisible = isVisible;

this.moons = moons;

this.lightyearsFromSun = lightyearsFromSun;

}

//getters & setters

public void set\_name(String name){

this.name = name;

}

public String get\_name(){

return name;

}

public void set\_size(double size){

this.size = size;

}

public double get\_size(){

return size;

}

public void set\_color(String color){

this.color = color;

}

public String get\_color(){

return color;

}

public void set\_isVisible(boolean isVisible){

this.isVisible = isVisible;

}

public boolean get\_isVisible(){

return isVisible;

}

public void set\_moons(int moons){

this.moons = moons;

}

public int get\_moons(){

return moons;

}

public void set\_lightyearsFromSun(double lightyearsFromSun){

this.lightyearsFromSun = lightyearsFromSun;

}

public double get\_lightyearsFromSun(){

return lightyearsFromSun;

}

//member methods

public void explode(){

int time = 10;

while(time-->0){

try{

Thread.sleep(1000);}

catch(InterruptedException ex){

Thread.currentThread().interrupt();}

System.out.println(name +" will self destruct in "+time+" seconds!");

}

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

size = 0.0;

}

public void closerToSun(double x){

if(lightyearsFromSun-x>0){

lightyearsFromSun = lightyearsFromSun - x;

System.out.println(name +" is "+lightyearsFromSun+ " lightyears from the sun");

}

else{

System.out.println("Warning, can't move that close to the Sun!");

}

}

public void changeColor(String color){

this.color = color;

System.out.println(name +" is now "+ color);

}

public void addMoons(int m){

moons = moons + m;

System.out.println("you added "+m+" moons to "+ name);

System.out.println("there are now "+moons+" moons");

}

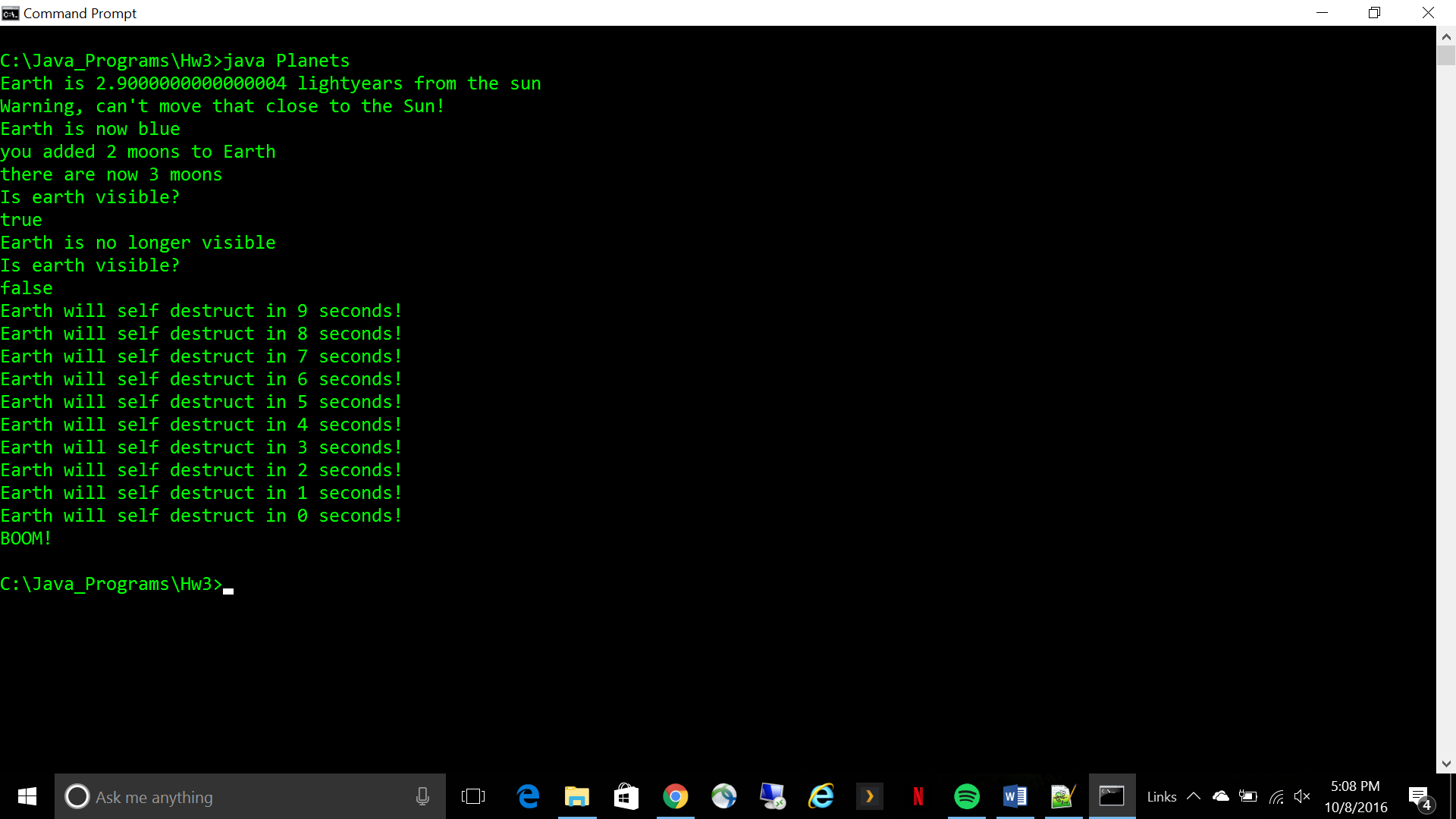
public void hide(){

isVisible = false;

System.out.println(name+" is no longer visible");

}

}



Format and write data into the next line of the output file

Take the pieces of desired data and put into substrings

No

Close the files

Yes

No

Last line of file?

Yes

Skip the line

Line contain desired data?

Read the next line in the input file

Input file

/\* Weslyn Wagner - zfs119

\* Sept8

\*

\* Version 1.0 2016/10/08

\*

\* EE 5343 UTSA

\*/

import java.io.\*;

import java.util.\*;

class sampleFile{

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

String infile = "C:\\JavaTextFiles\\SampleFile.txt";

String line;

int len;

String timeval;

String totval;

String conval;

String outfile = "C:\\JavaTextFiles\\OutFile.txt";

String write = "time, tot\_iter, conv\_iter";

try{

FileReader freader = new FileReader(infile);

BufferedReader in = new BufferedReader(freader);

FileWriter fwrite = new FileWriter(outfile);

PrintWriter out = new PrintWriter(fwrite);

//System.out.println("time,"+" "+"tot\_iter,"+" "+"conv\_iter");

out.println(write);

while((line = in.readLine()) != null){

if(line.contains("tran: time")){

line.trim();

len = line.length();

//there are 63 chars

timeval = line.substring(13,24);

totval = line.substring(39,42);

conval = line.substring(60);

out.println(timeval+", "+totval+", "+conval);

}

}

in.close();

out.close();

}

catch (IOException f){

System.out.println("Error "+ f);

}

System.out.println("file successfully created in: "+outfile);

}

}

