/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

MAIN PROGRAM

Author : Jonathan Parrilla

Course : COP 3804 MW 7:50 PM - 9:05PM

Professor : Michael Robinson

Program # : pgm1

- Implement a main program that will use external classes and constructors using

- Enhanced for loops

- Final Variables

- Variable-Length Argument List in Methods

- ternary "if"

- Use GUI messageDialog Boxes and inputDialogBoxes for all communications with the user.

- Each task must be done inside its own method.

- Use the main method to create variables and call the methods ONLY.

- Make the FINAL variables global.

- External class and Contructors DO NOT contain data, they PROCESS data sent by the user (main program)

Due Date : 09/12/2012

Certification:

I hereby certify that this work is my own and none of it is the work of any other person.

..........{ your signature }..........

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**import** javax.swing.JOptionPane;

**public** **class** ParrillaJpgm1

{

//Misc Methods

**public** **static** **void** p(Object stuff)

{

System.*out*.println(stuff);

}

**public** **static** **void** pl()

{

System.*out*.println();

}

**public** **static** **void** div()

{

*p*("--------------------------------------------------------");

}

//GUI Methods

**public** **static** String doInputBox(String question)

{

**return**(JOptionPane.*showInputDialog*(question));

}

**public** **static** **void** doMessageBox (Object message)

{

JOptionPane.*showMessageDialog*(**null**, message);

}

//Methods that will be called

**public** **static** **void** GetName()

{

*p*("Getting Name Method");

*pl*();

//Object of a class that accepts nothing

ParrillaJConst Nothing = **new** ParrillaJConst();

//Use this constructor's sets and gets to set the name and print it

Nothing.setFirstName(*doInputBox*("Enter your first name"));

Nothing.setLastName(*doInputBox*("Enter your last name"));

*doMessageBox*("Hello " +Nothing.getFirstName() +Nothing.getLastName());

*pl*();

*div*();

}

**public** **static** **void** NumberSorter()

{

*p*("Sorting Numbers Method");

*pl*();

**final** Object numbers[] = {66, 40, 2};

ParrillaJClass.*Sorter*(numbers);

*pl*();

*div*();

}

**public** **static** **void** TheGrades()

{

*p*("Average Grades Method");

*pl*();

**final** **double** Grades[] = {96,98,95,95,100,95,96,100};

*doMessageBox*("Your expected grades are:\n"

+"Program 1: " +Grades[0] +"\n"

+"Program 2: " +Grades[1] +"\n"

+"Program 3: " +Grades[2] +"\n"

+"Program 4: " +Grades[3] +"\n"

+" Exam 1: " +Grades[4] +"\n"

+" Exam 2: " +Grades[5] +"\n"

+" Exam 3: " +Grades[6] +"\n"

+"Final Exam: " +Grades[7]);

ParrillaJConst Averager = **new** ParrillaJConst(Grades);

*doMessageBox*("Your Average would be: " +ParrillaJClass.*GradeAverage*(Averager.getGrades()));

*pl*();

*div*();

}

**public** **static** **void** Information()

{

*p*("Getting Information Method");

**final** String major = *doInputBox*("What is your major?");;

**final** String creditsTaken = *doInputBox*("How many credits have you taken?");;

**final** String creditsNeeded = *doInputBox*("How many credits do you need to graduate?");;

**final** String currentGPA = *doInputBox*("What is your current GPA?");;

ParrillaJClass.*MyMajor*(major);

ParrillaJClass.*MyCreditsTaken*(creditsTaken);

ParrillaJClass.*MyCreditsNeeded*((creditsNeeded));

ParrillaJClass.*MyGPA*(currentGPA);

}

//Main Method

**public** **static** **void** main (String arg[])

{

*p*("Starting the program.");

*pl*();

*div*();

//call a method that gets your name

*GetName*();

//call a method that accepts three numbers as array of objects

*NumberSorter*();

//call a method that accept grades as an array of objects.

*TheGrades*();

//Calling methods that will accept the following variables

*Information*();

*pl*();

*p*("The End");

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

EXTERNAL CLASS

Author : Jonathan Parrilla

Course : COP 3804 MW 7:50 PM - 9:05PM

Professor : Michael Robinson

Program # : pgm1

- Implement a main program that will use external classes and constructors using

- Enhanced for loops

- Final Variables

- Variable-Length Argument List in Methods

- ternary "if"

- Use GUI messageDialog Boxes and inputDialogBoxes for all communications with the user.

- Each task must be done inside its own method.

- Use the main method to create variables and call the methods ONLY.

- Make the FINAL variables global.

- External class and Contructors DO NOT contain data, they PROCESS data sent by the user (main program)

Due Date : 09/12/2012

Certification:

I hereby certify that this work is my own and none of it is the work of any other person.

..........{ your signature }..........

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**import** javax.swing.JOptionPane;

**public** **class** ParrillaJClass

{

//Misc Methods

**public** **static** **void** p(Object stuff)

{

System.*out*.println(stuff);

}

**public** **static** **void** pl()

{

System.*out*.println();

}

**public** **static** **void** div()

{

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

*p*("--------------------------------------------------------");

}

//GUI Methods

**public** **static** **void** doMessageBox (Object message)

{

JOptionPane.*showMessageDialog*(**null**, message);

}

**public** **static** **double** GradeAverage(**double** grades[])

{

**double** total = 0;

**for**(**double** temp: grades)

{

total = total + temp;

}

**return** (total/grades.length);

}

**public** **static** **void** Sorter(Object Numbers[])

{

**int** fn = (**int**)Numbers[0];

**int** sn = (**int**)Numbers[1];

**int** tn = (**int**)Numbers[2];

ParrillaJConst SortNumbers = **new** ParrillaJConst (fn, sn, tn);

*doMessageBox*("The Numbers 66, 40, and 2 in ascending order are: \n"

+ SortNumbers.getfirstNumber() + " , "

+ SortNumbers.getsecondNumber() + " , "

+ SortNumbers.getthirdNumber());

}

**public** **static** **void** MyMajor( String major)

{

//print my major

*doMessageBox*("Your Major is: " +major);

}

**public** **static** **void** MyCreditsTaken(String ct)

{

//Print credits taken

*doMessageBox*("Your credits taken are: " +ct);

}

**public** **static** **void** MyCreditsNeeded(String cn)

{

//print credits needed to graduate

*doMessageBox*("Your credits needed to graduate are: " +cn);

}

**public** **static** **void** MyGPA(String gpa)

{

//print current gpa

*doMessageBox*("Your current GPA is: " +gpa);

}

}

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

CONSTRUCTOR

Author : Jonathan Parrilla

Course : COP 3804 MW 7:50 PM - 9:05PM

Professor : Michael Robinson

Program # : pgm1

- Implement a main program that will use external classes and constructors using

- Enhanced for loops

- Final Variables

- Variable-Length Argument List in Methods

- ternary "if"

- Use GUI messageDialog Boxes and inputDialogBoxes for all communications with the user.

- Each task must be done inside its own method.

- Use the main method to create variables and call the methods ONLY.

- Make the FINAL variables global.

- External class and Contructors DO NOT contain data, they PROCESS data sent by the user (main program)

Due Date : 09/12/2012

Certification:

I hereby certify that this work is my own and none of it is the work of any other person.

..........{ your signature }..........

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

**public** **class** ParrillaJConst

{

//private variables

**private** **int** firstNumber;

**private** **int** secondNumber;

**private** **int** thirdNumber;

//private double Grades[] = {0,0,0,0,0,0,0,0};

**private** **double** Grades[] = **new** **double**[8];

/\*private double Grades[];

\* Caused a java.lang.NullPointerException in Main,

\* which was tied to line 26 of this constructor,

\* Which is where I tried to initialize Grades[],

\* using an enhanced for loop.

\*/

**private** String FirstName;

**private** String LastName;

//constructors

**public** ParrillaJConst()

{

}

**public** ParrillaJConst(**double** ClassGrades[])

{

**int** x = 0;

**for**(**double** temp: ClassGrades)

{

Grades[x] = temp;

System.*out*.println("Grades: " +Grades[x]);

x++;

}

}

**public** ParrillaJConst(**int** fn, **int** sn, **int** tn)

{

firstNumber = fn;

secondNumber = sn;

thirdNumber = tn;

//Sort the numbers in ascending order

**int** temp = 0;

**if**(firstNumber > secondNumber)

{

temp = secondNumber;

secondNumber = firstNumber;

firstNumber = temp;

}

**if**(secondNumber > thirdNumber)

{

temp = thirdNumber;

thirdNumber = secondNumber;

secondNumber = temp;

}

**if**(firstNumber > secondNumber)

{

temp = secondNumber;

secondNumber = firstNumber;

firstNumber = temp;

}

}

//Setters

**public** **void** setfirstNumber(**int** fn)

{

firstNumber = fn;

}

**public** **void** setsecondNumber(**int** sn)

{

secondNumber = sn;

}

**public** **void** setthirdNumber(**int** tn)

{

thirdNumber = tn;

}

**public** **void** setFirstName(String name)

{

FirstName = name;

}

**public** **void** setLastName(String name)

{

LastName = name;

}

**public** **void** setGrades(**double**[] rGrades)

{

Grades[0] = rGrades[0];

Grades[1] = rGrades[1];

Grades[2] = rGrades[2];

Grades[3] = rGrades[3];

}

//Getters

**public** String getLastName()

{

**return** LastName;

}

**public** String getFirstName()

{

**return** FirstName;

}

**public** **double**[] getGrades()

{

**return** Grades;

}

**public** **int** getfirstNumber()

{

**return** firstNumber;

}

**public** **int** getsecondNumber()

{

**return** secondNumber;

}

**public** **int** getthirdNumber()

{

**return** thirdNumber;

}

}