Daron Adkins

JAVA Programming

Project 2

**PROBLEM STATEMENT**

Issue:

Your client, Technologies R Us, needs a program to bill their clients. The billing is based on the team member’s title and the number of hours worked.

|  |  |
| --- | --- |
| **Title** | **Hourly Rate** |
| Documentation Specialist | $25.00 |
| Quality Assurance Specialist | $35.00 |
| Analyst | $75.00 |
| Software Engineer | $75.00 |

Project Specifications:

1. The program allows for multiple clients invoices to be entered and multiple team members per client.
2. The user enters the client’s name.
3. The user enters the team member’s name and title.
4. The program calculates the total amount billed by team member and overall total.
5. The program stores the client’s name, invoice number (randomly generated), team member’s name, team member’s billing amount, and overall client amount in a data file.
6. The program reads the data file created and displays the client’s name, invoice number, team member’s name, team member’s billing amount, and overall client amount.
7. The program must utilize at least two methods that pass data.
8. All entered data must be validated – no bad data!!

**ALGORITHM**

1. Get file name for invoice
2. Get company name for whom invoice is being made
3. Get number of employees that worked for client
4. Get Employee’s name
5. Get Employee’s job title
6. Calculate hourly rate based on job title
7. Get number of hours employee worked
8. Calculate total amount employee earned
9. Repeat 4 – 8 for each employee
10. Write data to file
11. Read data from file

**IPO CHART**

|  |  |  |
| --- | --- | --- |
| **Input** | **Process** | **Output** |
| fileName | employeeTotal = hoursWorked \* hourlyRate; | fileName |
| companyName | grandTotal += employeeTotal | companyName |
| numEmployees |  | numEmployees |
| employeeName |  | employeeName |
| jobTitle |  | hoursWorked |
| hoursWorked |  | hourlyRate |
|  |  | jobTitle |
|  |  | employeeTotal |
|  |  | grandTotal |

**VARIABLES**

String input;

String fileName

String companyName

String employeeName

String jobTitle

int numEmployees

int reply

int keepGoing

int flag

int invoiceNumber;

float hourlyRate

float hoursWorked

float employeeTotal

float grandTotal

**FORMULAS**

employeeTotal = hoursWorked \* hourlyRate

grandTotal += employeeTotal

**HIERARCHY**

**PSEUDOCODE**

**void main()**

//begin main module

//declarations and initilizations

Declare keepGoing As Int = 0

While (keepGoing == 0)

Declare filename As String

Declare companyName As String

Declare employeeName As String

Declare jobTitle As String

Declare reply As String

Declare numEmployees As Float = 0.0

Declare counter As Int = 0

Declare flag As Int = 0

Declare hourlyRate As Float = 0.0

Declare hoursWorked As Float = 0.0

Declare employeeTotal As Float = 0.0

Declare grandTotal As Float = 0.0

Write “Welcome to Invoice Pro! This program will let you create

Multiple invoices with multiple team members per invoice.”

Write “Please enter the name for your invoice file:”

Input fileName

Open File “filename” for Output as outputFile

Write “Please enter the name of the company for whom you are writing the invoice”

Input companyName

Write companyName to outputFile

While (flag == 0)

Write “How many employees worked for the client?”

Input numEmployees

If(dataValidation(numEmployees) == true)

flag = 1

else

Write “Please enter a number greater than 0”

End If

End While

for(counter = 1; counter <= numEmployees; counter++)

flag = 0

Write “Please enter the name of employee #” +counter +”: “

Input employeeName

Write employeeName to outputFile

Write “What is their job title?”

Input jobTitle

hourlyRate = hourlyRateMethod(jobTitle)

Write jobTitle to outputFile

Write hourlyRate to outputFile

While (flag == 0)

Write “How many hours did the employee work?”

Input hoursWorked

If(dataValidation(hoursWorked) == true)

flag = 1

else

Write “Please enter a number greater than 0”

End If

End While

Write hoursWorked to outputFile

employeeTotal = hoursWorked \* hourlyRate

Write employeeTotal to outputFile

End For

grandTotal = grandTotal + employeeTotal

Write grandTotal to outputFile

Write “The data has been written to the file.”

Close outputFile

//now let’s read the file

Open File “filename” for Input as inputFile

While (inputFile hasNext)

Write inputFile

End while

Close inputFile

Write “Would you like to enter another invoice?”

Input reply

If (reply == “yes”)

keepGoing = 0

else

keepGoing = 1

End

//end of main module

**float hourlyRateMethod(String hrTitle)**

//this module just assigns an hourly rate

if (hrTitle == “Analyst”)

return 25.00

else if(hrTitle == “Documentation Specialist”)

return 35.00

else

return 75.00

//end of module

**Boolean dataValidation(float var)**

//validates our data

If (var > 0)

Return true

Else

Return false

//end of module

**TEST DATA – 5 complete data sets**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Input** | **Process** | **Output** |
| fileName | Test 1 | 10 \* 25 = 250  0 + 250 = 250 | Filename: Test 1 |
| companyName | Lowes | Company Name: Lowes |
| numEmployees | 1 | Employee Name: Daron Adkins |
| employeeName | Daron Adkins | Job Title: Analyst |
| Hourly Rate: $25.00 |
| Hours worked: 10 |
| jobTitle | Analyst | Rate total: $250 |
| hoursWorked | 10 |  | Grand Total: $250 |
|  |  |  |  |
|  |  |  |  |
| fileName | Test 2 | 10 \* 25 = 250  10 \* 75 = 750  750+ 250 = 250 | Filename: Test 2 |
| companyName | Hardees | Company Name: Hardees |
| numEmployees | 2 | Employee Name: Daron Adkins, John Doe |
| employeeName | Daron Adkins  John Doe | Job Title: Analyst, Software Engineer |
| Hourly Rate: $25.00, $75.00 |
| Hours worked: 10, 10 |
| jobTitle | Analyst, Software Engineer | Rate total: $250, $750 |
| hoursWorked | 10, 10 |  | Grand Total: $1000 |
|  |  |  |
| fileName | Test 3 | 10 \* 35 = 250  0 + 350 = 350 | Filename: Test 3 |
| companyName | Acme | Company Name: Acme |
| numEmployees | 1 | Employee Name: Daron Adkins |
| employeeName | Daron Adkins | Job Title: Documentation Specialist |
| Hourly Rate: $35.00 |
| Hours worked: 10 |
| jobTitle | Documentation Specialist | Rate total: $350 |
| hoursWorked | 10 |  | Grand Total: $350 |
|  |  |  |  |
|  |  |  |  |
| fileName | Test 4 | 1 \* 75 = 75  0 + 75 = 75 | Filename: Test 4 |
| companyName | GCSC | Company Name: GCSC |
| numEmployees | 1 | Employee Name: John Doe |
| employeeName | John Doe | Job Title: Quality Assurance Specialist |
| Hourly Rate: $75.00 |
| Hours worked: 1 |
| jobTitle | Quality Assurance Speialist | Rate total: $75 |
| hoursWorked | 1 |  | Grand Total: $75 |
|  |  |  |  |
| fileName | Test 5 | 10\* 25 = 250  10\* 25 = 250  10\* 25 = 250  250 + 250 + 250 = 750 | Filename: Test 5 |
| companyName | My Dad | Company Name: My Dad |
| numEmployees | 3 | Employee Name: Daron Adkins, John Doe, Jane Doe |
| employeeName | Daron Adkins  John Doe  Jane Doe | Job Title: Analyst, Analyst, Analyst |
| Hourly Rate: $25.00 |
| Hours worked: 10, 10 10 |
| jobTitle | Analyst, Analyst, Analyst | Rate total: $250, $250, $250 |
| hoursWorked | 10, 10, 10 |  | Grand Total: $750 |