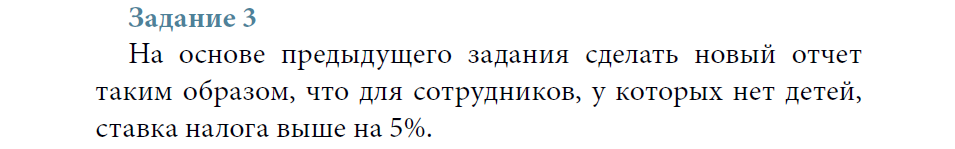
Домашнее задание № 7.2

ТЕМА: НАСЛЕДОВАНИЕ

******

***class Workers3 extends Workers2***

package com.company.task3;  
  
import com.company.task2.Workers2;  
  
//Класс "Работники" - Workers3 наследует от Workers2  
public class Workers3 extends Workers2 *{* //поля  
 protected boolean children;// дети у сотрудника  
 protected int taxChild = 5; // добавляется налог 5%, для работников без детей  
  
 // конструкторы  
 // 1.конструктор с 5 параметрами  
 public Workers3*(*String *surname*, String *initials*, String *typeOfPayment*,  
 double *time*, double *payment) {* super*(surname*, *initials*, *typeOfPayment*, *time*, *payment)*;  
 *}* // 2.конструктор с 6 параметрами  
 public Workers3*(*String *surname*, String *initials*, String *typeOfPayment*,  
 double *time*, double *payment*, boolean *children) {* super*(surname*, *initials*, *typeOfPayment*, *time*, *payment)*;  
 this.children = *children*;  
 *}* // getters and setters  
 public boolean isChildren*() {* return children;  
 *}* public void setChildren*(*boolean *children) {* this.children = *children*;  
 *}* public int getTaxChild*() {* return taxChild;  
 *}* public void setTaxChild*(*int *taxChild) {* this.taxChild = *taxChild*;  
 *}  
  
}*

***class WorkersRateOfPayment3 extends Workers3***

package com.company.task3;  
  
// класс WorkersRateOfPayment3 (ставка) - наследник Workers3  
  
public class WorkersRateOfPayment3 extends Workers3 *{* // конструкторы  
 // 1. конструктор с 5 параметрами  
 public WorkersRateOfPayment3*(*String *surname*, String *initials*, String *typeOfPayment*, double *rateDays*, double *daysWorked) {* super*(surname*, *initials*, *typeOfPayment*, *rateDays*, *daysWorked)*;  
 *}* // 2. конструктор с 6 параметрами  
 public WorkersRateOfPayment3*(*String *surname*, String *initials*, String *typeOfPayment*, double *rateDays*, double *daysWorked*, boolean *children) {* super*(surname*, *initials*, *typeOfPayment*, *rateDays*, *daysWorked*, *children)*;  
  
 *}* // методы  
 // 1. метод представления класса в виде строки - перегрузка метода toString  
 @Override  
 public String toString*() {* if *(*super.children*) {* return "|" + super.toString*()* + "|" + tax + "%|" + salaryCalculation*()* + " руб.|" +  
 taxRate*()* + " руб.|";  
 *}* else return "|" + super.toString*()* + "|" + *(*taxChild + tax*)* + "%|" + salaryCalculation*()* + " руб.|" +  
 taxRate*()* + " руб.|";  
 *}* // 2. метод вычисления заработной платы работника с учётом налога 20%  
 public double taxRate*() {* if *(*super.children*) {* return *(*salaryCalculation*()* - *(*salaryCalculation*()* \* tax / 100*))*;  
 *}* else  
 return *(*salaryCalculation*()* - *(*salaryCalculation*()* \* *(*taxChild + tax*)* / 100*))*;  
 *}  
}*

***class WorkersHourOfPayment3 extends Workers3***

package com.company.task3;  
  
// класс WorkersHourOfPayment3 (почасовая ставка) - наследник Workers3  
public class WorkersHourOfPayment3 extends Workers3 *{* // конструкторы  
 // 1. конструктор с 5 параметрами  
 public WorkersHourOfPayment3*(*String *surname*, String *initials*, String *typeOfPayment*,  
 double *rateHour*, double *hourWorked) {* super*(surname*, *initials*, *typeOfPayment*, *rateHour*, *hourWorked)*;  
 *}* // 2. конструктор с 6 параметрами  
 public WorkersHourOfPayment3*(*String *surname*, String *initials*, String *typeOfPayment*, double *rateHour*, double *hourWorked*, boolean *children) {* super*(surname*, *initials*, *typeOfPayment*, *rateHour*, *hourWorked*, *children)*;  
  
 *}* // методы  
 // 1. метод представления класса в виде строки - перегрузка метода toString  
 @Override  
 public String toString*() {* if *(*super.children*) {* return "|" + super.toString*()* + "|" + tax + "%|" + salaryCalculation*()* + " руб.|" +  
 taxOnHourly*()* + " руб.|";  
 *}* else return "|" + super.toString*()* + "|" + *(*taxChild + tax*)* + "%|" + salaryCalculation*()* + " руб.|" +  
 taxOnHourly*()* + " руб.|";  
 *}* // 2. метод вычисления заработной платы работника с учётом налога 20%  
 public double taxOnHourly*() {* if *(*super.children*) {* return *(*salaryCalculation*()* - *(*salaryCalculation*()* \* tax / 100*))*;  
 *}* else  
 return *(*salaryCalculation*()* - *(*salaryCalculation*()* \* *(*taxChild + tax*)* / 100*))*;  
 *}  
}*

***class WorkersPieceOfPayment3 extends Workers3***

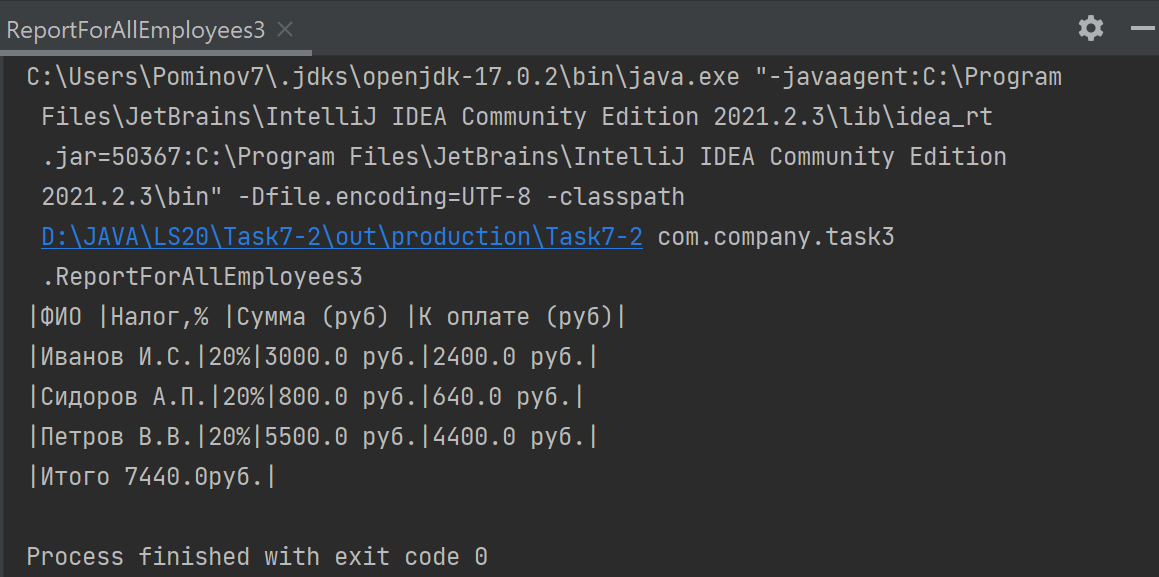
package com.company.task3;  
  
// класс WorkersPieceOfPayment3 (сдельная оплата) - наследник Workers3  
  
public class WorkersPieceOfPayment3 extends Workers3 *{* // конструкторы  
 // 1. конструктор с 5 параметрами  
 public WorkersPieceOfPayment3*(*String *surname*, String *initials*, String *typeOfPayment*,  
 double *piecework*, double *sumWorked) {* super*(surname*, *initials*, *typeOfPayment*, *piecework*, *sumWorked)*;  
  
 *}* // 2. конструктор с 6 параметрами  
 public WorkersPieceOfPayment3*(*String *surname*, String *initials*, String *typeOfPayment*,  
 double *piecework*, double *sumWorked*, boolean *children) {* super*(surname*, *initials*, *typeOfPayment*, *piecework*, *sumWorked*, *children)*;  
 *}* // методы  
 // 1. метод представления класса в виде строки - перегрузка метода toString  
 @Override  
 public String toString*() {* if *(*super.children*) {* return "|" + super.toString*()* + "|" + taxPiece + "%|" + salaryCalculation*()* + " руб.|" +  
 taxFix*()* + " руб.|";  
 *}* else return "|" + super.toString*()* + "|" + *(*taxChild + taxPiece*)* + "%|" + salaryCalculation*()* + " руб.|" +  
 taxFix*()* + " руб.|";  
 *}* // 2. метод вычисления заработной платы работника с учётом налога 15%  
 public double taxFix*() {* if *(*super.children*) {* return *(*salaryCalculation*()* - *(*salaryCalculation*()* \* taxPiece / 100*))*;  
 *}* else  
 return *(*salaryCalculation*()* - *(*salaryCalculation*()* \* *(*taxChild + taxPiece*)* / 100*))*;  
 *}  
}*

***class ReportForAllEmployees3***

package com.company.task3;  
  
public class ReportForAllEmployees3 *{* public static void main*(*String*[] args) {* //Создаём объект работника с видом оплаты труда "Ставка"  
 WorkersRateOfPayment3 *employee1* = new WorkersRateOfPayment3*(*"Иванов", "И.С.", "Ставка",  
 100, 30, true*)*;// у этого сотрудника есть дети  
  
 //Создаём объект работника с видом оплаты труда "Почасовая"  
 WorkersHourOfPayment3 *employee2* = new WorkersHourOfPayment3*(*"Сидоров", "А.П.", "Почасовая", 100, 8, true*)*;// у этого сотрудника есть дети  
  
 //Создаём объект работника с видом оплаты труда "Сдельная"  
 WorkersPieceOfPayment3 *employee3* = new WorkersPieceOfPayment3*(*"Петров", "В.В.", "Сдельная", 550, 10*)*;// у этого сотрудника нет детей, ставка налога станет на 5% больше  
  
 //вывод в консоль отчета по всем сотрудникам фирмы о заработной плате  
 System.*out*.println*(*"|ФИО " + "|Налог,% " + "|Сумма (руб) |" + "К оплате (руб)|"*)*;  
 System.*out*.println*(employee1)*;  
 System.*out*.println*(employee2)*;  
 System.*out*.println*(employee3)*;  
 System.*out*.println*(*"|Итого " + *(employee1*.taxRate*()* + *employee2*.taxOnHourly*()* +  
 *employee3*.taxFix*())* + "руб.|"*)*;  
 *}  
}*

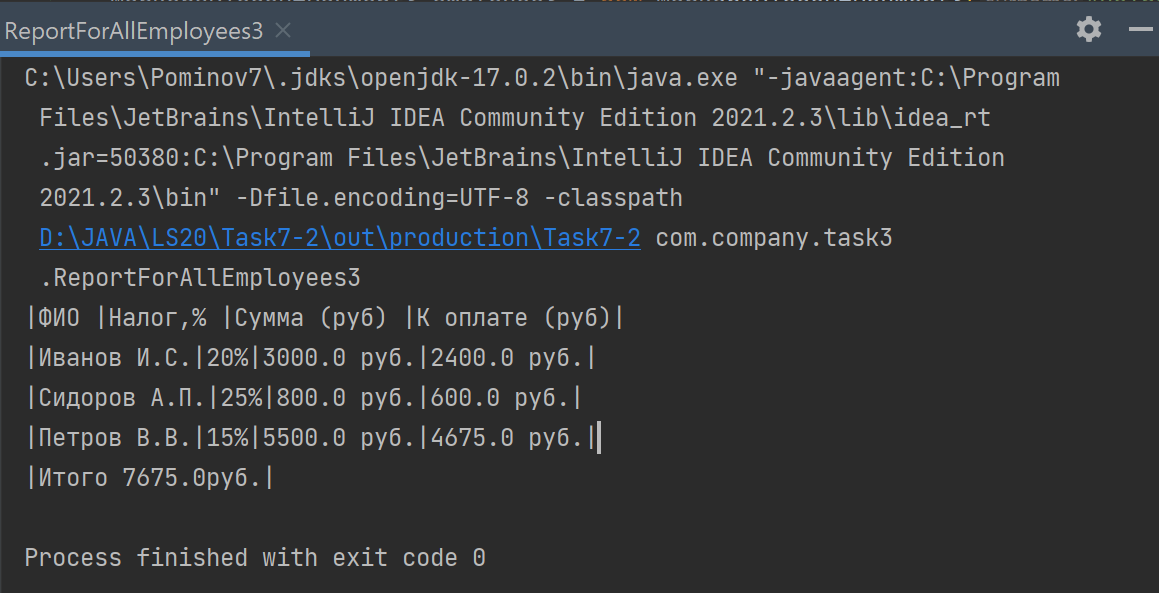
***Тест***

***(у Петрова В.В. нет детей):***

******

***Тест***

***(у Петрова В.В. родился ребенок, а у Сидорова А.П. ребенок стал совершеннолетним):***

******