***Система за Съхранение на Информация и Обработка на Данни на Служители***

Име: Виктор Методиев

Фак. Номер: 361223016

Факултет: ФИТ

Специалност: ИСИИ

Група: 21

Supervisor: Веска Ганчева

**Описание на използваните модули (класове) – елементи и предназначение:**

Първият дефиниран клас в програмата е Person. Kласът има за функция да създаде person с private полета за име, адрес и id. Класът съдържа и public методи, които служат за getters и setters и конструктор, приемащ име, адрес и id. Тези методи задават стойности и да се достъпват. Класът има и функция за показване на информацията на човека както и деструктор.

Следващият дефиниран клас е Employee, който наследява класът Person. Класът Employee има за задача да създаде employee, като той взима името, адресът и id-то, и допълва чрез отработени часове и часово заплащане като private стойностите. В този клас също има getters и setters. Има както default constructor, така и конструктор, който приема име, адрес, id, отработени часове и часово заплащане. Класът създава служител и смята неговите доходите спрямо неговите отработени часове умножени по часовото му разплащане. Смятането на доходите се случва в приятелска функция. Има и деструктор, който заличава служителя след неговото създаване.

В main метода се случва добавянето на брой на служители, както и въвеждането на техните данни. Данните се запазват в масив и се подават на класа Employee, където се случва и тяхната обработка.

**Общо описание за функциониране на програмата (вход/изход)**:

С включването на програмата потребителят има възможност да въведе число между 0 и 50, ако числото не е в тоза граница, на потребителя се налага да въведе отново число, докато число не е в правилния диапазон, а имено от 0 до 50. Въведеното число съответства на броя на служителите. След това се пристъпва към попълването на данните на всеки член от фирмата. Данните са: име от тип string, адрес от тип string, id от тип int, заработени часове от тип int и часова ставка от тип double. След въвеждането на даннинте на всички членове програмата изчислява и изкарва на конзолата колко пари е заработил всеки служител, както и средните заработени пари взимайки доходите на всички членове и разделяйки ги на техния брой.

**Листинг на изходния код на програмата:**

#include <iostream>

using namespace std;

class Person {

    private:

        string name;

        string address;

        int ID;

    public:

        void setName(string n) { name = n; };

        void setAddress(string adr) { address = adr; };

        void setID(int id) { ID = id; };

        string getName() { return name; };

        string getAddress() { return address; };

        int getID() { return ID; };

        Person(string n, string adr, int id);

        void displayPerson();

    ~Person()

    {

        cout << "Person saved and destroyed!" << endl;

    }

};

Person::Person(string n, string adr, int id)

{

    name = n;

    address = adr;

    ID = id;

};

void Person::displayPerson()

{

    cout << "Name: " << name << endl;

    cout << "Address: " << address << endl;

    cout << "ID: " << ID << endl;

};

class Employee : public Person {

    private:

        int workedHours;

        double hourlyRate;

    public:

        void setWorkedHours(int hours) { workedHours = hours; };

        void setHourlyRate(double rate) { hourlyRate = rate; };

        int getWorkedHours() { return workedHours; };

        double getHourlyRate() { return hourlyRate; };

        Employee() : Person("", "", 0), workedHours(0), hourlyRate(0.0) {};

        Employee(string n, string adr, int id, int hours, double rate);

        void displayEmployee();

*friend* double monthlyPayment(Employee*&* y);

    ~Employee()

    {

        cout << "Employee saved and destroyed!" << endl;

    }

};

Employee::Employee(string n, string adr, int id, int hours, double rate) : Person(n, adr, id)

{

    workedHours = hours;

    hourlyRate = rate;

};

void Employee::displayEmployee()

{

    displayPerson();

    cout << "Worked hours this month: " << workedHours << endl;

    cout << "Hourly rate: " << hourlyRate << endl;

};

double monthlyPayment(Employee*&* y) {

    double salary = y.workedHours \* y.hourlyRate;

    return salary;

};

int main ()

{

    string n, adr;

    int id, hours;

    double rate;

    int numberOfEmployees;

    do

    {

        cout << "Enter number for employees(between 0 and 50): ";

        cin >> numberOfEmployees;

    } while (numberOfEmployees <= 0 || numberOfEmployees > 50);

    Employee \*arr = new Employee[numberOfEmployees];

    for (int i = 0; i < numberOfEmployees; i++)

    {

        cout << "Enter name: ";

        cin >> n;

        cout << "Enter address: ";

        cin >> adr;

        cout << "Enter ID: ";

        cin >> id;

        cout << "Enter worked hours: ";

        cin >> hours;

        cout << "Enter hourly rate: ";

        cin >> rate;

        arr[i] = Employee(n, adr, id, hours, rate);

    }

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

        arr[i].displayEmployee();

    }

    cout << "Employees monthly payments: " << endl;

    for (int i = 0; i < numberOfEmployees; i++)

    {

        cout << arr[i].getName() << " earned " << monthlyPayment(arr[i]) << "$ this month." << endl;

    }

    double sum = 0;

    for (int i = 0; i < numberOfEmployees; i++)

    {

        sum += monthlyPayment(arr[i]);

    }

    double avgPayment = sum / numberOfEmployees;

    cout << "The Average monthly payment of employees is " << avgPayment << "$." << endl;

    delete[] arr;

    return 0;

};

**Резултати от имплементацията на програмата:**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**