# Міністерство освіти і науки України

Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського"

# Факультет інформатики та обчислювальної техніки Кафедра інформатики та програмної інженерії

Звіт

з лабораторної роботи № 9 з дисципліни

«Основи програмування

»

Варіант 23

Виконав студент Панченко Сергій Віталійович

(шифр, прізвище, ім'я, по батькові)

Перевірив

( прізвище, ім'я, по батькові)

Київ 202

## Лабораторна робота 3

**Мета:** визначити механізми створення класів та об’єктів

**Постановка задачі:**

**Розробити клас «працівник», який характеризується ПІБ, датою народження( ММ-ДД-РРРР ). Створити масив об’єктів. Визначити найстаршого працівника.**

C:

#include <iostream>

#include"Header.h"

using namespace std;

class Worker {

private:

string patronym;

string name;

string surname;

string birth\_date;

public:

Worker();

Worker(string patronym, string name, string surname, string birth\_date);

string GetPatronym();

string GetName();

string GetSurname();

string GetBirthDate();

void SetPatronym(string patronym);

void SetName(string name);

void SetSurname(string surname);

void SetBirthDate(string birth\_date);

static bool WorkerAgeComparator(Worker w1, Worker w2);

};

class Company {

private:

vector<Worker>workers;

public:

Company();

Worker get\_oldest();

};

int main()

{

Company c = Company();

Worker old\_worker = c.get\_oldest();

cout << "Surname: " << old\_worker.GetSurname()<<"\t"

<< "Name: " << old\_worker.GetName()<<"\t"

<< "Patronym: " << old\_worker.GetPatronym()<<"\t"

<< "BirthDate: " << old\_worker.GetBirthDate() << endl;

system("pause");

return 0;

}

Worker::Worker() {};

Worker::Worker(string patronym, string name, string surname, string birth\_date) {

this->patronym = patronym;

this->name = name;

this->surname = surname;

this->birth\_date = birth\_date;

}

string Worker::GetPatronym() {

return this->patronym;

}

string Worker::GetName() {

return this->name;

}

string Worker::GetSurname() {

return this->surname;

}

string Worker::GetBirthDate() {

return this->birth\_date;

}

void Worker::SetPatronym(string patronym) {

this->patronym = patronym;

}

void Worker::SetName(string name) {

this->name = name;

}

void Worker::SetSurname(string surname) {

this->surname = surname;

}

void Worker::SetBirthDate(string birth\_date) {

this->birth\_date = birth\_date;

}

bool Worker::WorkerAgeComparator(Worker w1, Worker w2) {

string age1;

string age2;

for (int i = 0; i < w1.birth\_date.length(); i++)

{

if (isdigit(w1.birth\_date[i])) {

age1 += w1.birth\_date[i];

age2 += w2.birth\_date[i];

}

}

string age1\_c;

string age2\_c;

for (int i = 4; i < 8; i++)

{

age1\_c += age1[i];

age2\_c += age2[i];

}

for (int i = 0; i < 4; i++)

{

age1\_c += age1[i];

age2\_c += age2[i];

}

return atoi(age1\_c.c\_str()) < atoi(age2\_c.c\_str());

}

Company::Company() {

string buf = "y";

int count = 1;

while (buf == "y") {

string patronym;

string name;

string surname;

string birth\_date;

cout << "Worker " << count << endl;

cout << "Patronym: ";

cin >> patronym;

cin.ignore();

cout << "Name: ";

cin >> name;

cin.ignore();

cout << "Surname: ";

cin >> surname;

cin.ignore();

cout << "Date of Birth( MM-DD-YYYY ): ";

cin >> birth\_date;

cin.ignore();

this->workers.push\_back(

Worker(patronym, name, surname, birth\_date)

);

do {

cout << "Wann`a continue?y/n: " << endl;

cin >> buf;

cin.ignore();

} while (buf != "y" && buf != "n");

count++;

}

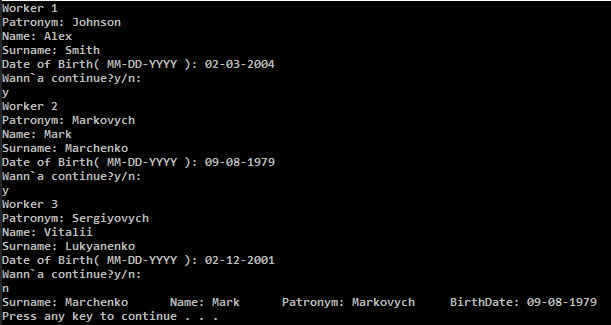
}

Worker Company::get\_oldest() {

return \*min\_element(this->workers.begin(), this->workers.end(), Worker::WorkerAgeComparator);

}

**Результат:**

****

**Висновок:** під час лабораторної роботи навчилися працювати з класами та об’єктами. Оскільки програма створюють об’єкти класу та визначає правильно найстаршого працівника, виконуючи умови поставленої до варіанту задачі, то лабораторна виконана правильно.