Федеральное государственное образовательное учреждение

Высшего профессионального образования

«Ярославский Государственный Технический Университет»

Кафедра «Информационные системы и технологии»

Работа защищена

с оценкой \_\_\_\_\_\_\_\_

Преподаватель

\_\_\_\_\_\_\_\_ Безрукова Е.Г.

«\_\_\_» \_\_\_\_\_\_\_\_\_ 2024

**Структуры**

Отчет о лабораторной работе №1

По дисциплине «Объектно-ориентированное программирование»

ЯГТУ 09.03.04 – 006 ЛР

Работу выполнил

студент группы

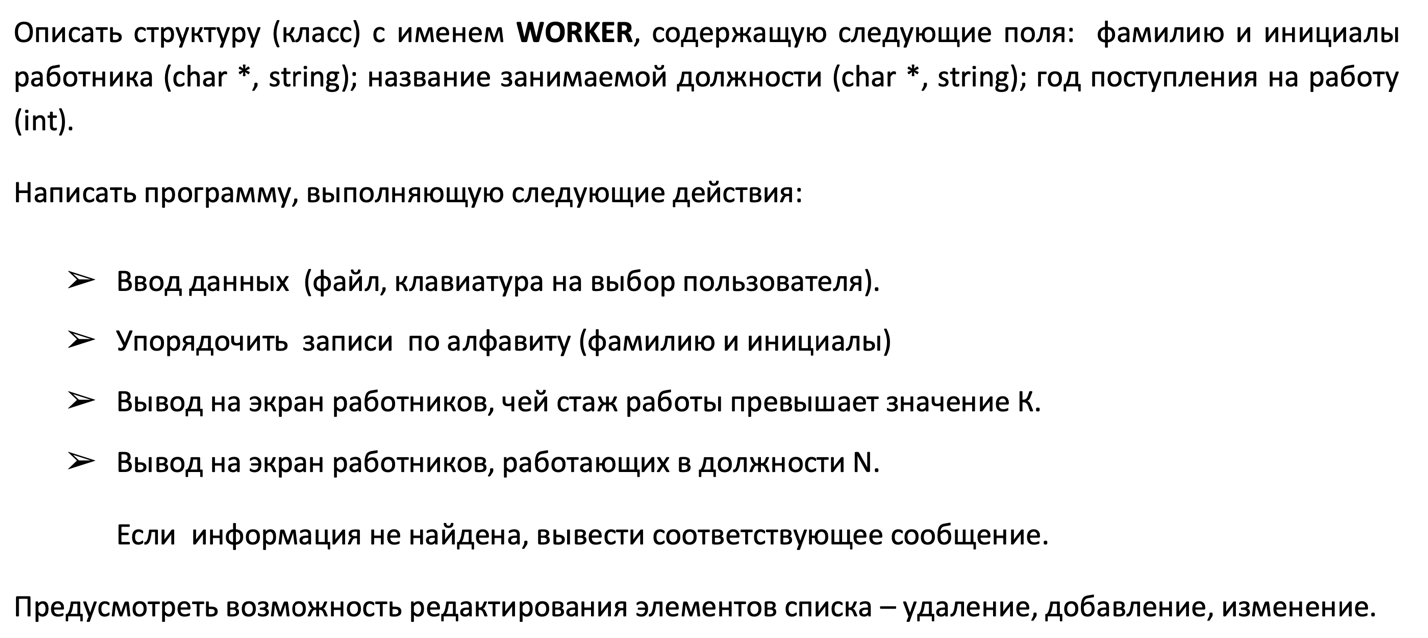
ЦПИ-11

Ушаков П.А.

«15» апреля 2024

Ярославль 2024

Задание:



Код:

Worker.h

#ifndef ***WORKER\_H***

#define ***WORKER\_H***

#include <iostream>

#include <string>

#include "exception.h"

***// класс Worker - информация о работнике***

struct ***Worker*** {

***// поля класса***

std::***string*** name;

std::***string*** job;

int year;

public:

***// оператор сравнения для сортировки***

bool ***operator<***(const ***Worker***& other) const {

int len = std::***min***(name.***length***(), other.name.***length***());

int i = 0;

while (i < len && std::***toupper***(name***[***i***]***) == std::***toupper***(other.name***[***i***]***)) {

i++;

}

return std::***toupper***(name***[***i***]***) < std::***toupper***(other.name***[***i***]***);

}

***// вывод информации***

void ***printInfo***() const {

std::cout ***<<*** "\x1b[32mИмя:\x1b[32m " ***<<*** name ***<<*** "\n"

***<<*** "Должность: " ***<<*** job ***<<*** "\n"

***<<*** "Год принятия на работу: " ***<<*** year ***<<*** "\n";

std::cout ***<<*** "\x1b[0m";

std::cout ***<<*** "\n";

}

***// геттеры***

std::***string*** ***getName***() const {return name;}

std::***string*** ***getJob***() const {return job;}

int ***getYear***() const {return year;}

***// сеттеры***

void ***setName***(const std::***string***& name) { this->name ***=*** name; }

void ***setJob***(const std::***string***& job) { this->job ***=*** job; }

void ***setYear***(int year) { this->year = year; }

};

#endif

Exception.h

#ifndef ***EXCEPTION\_H***

#define ***EXCEPTION\_H***

#include <iostream>

#include <string>

class ***TypeError*** {

public:

***TypeError***() {}

std::***string*** ***what***() {

return message;

}

private:

std::***string*** message = "\x1b[31mTypeError! Expected int\x1b[0m\n";

};

#endif

Workers.h

#include "worker.h"

#include <vector>

#include <fstream>

#include <sstream>

#include <algorithm>

#include <limits>

***// класс Workers - управление информацией о работниках***

struct ***Workers*** {

***// вектор с информацией о работниках***

std::***vector***<***Worker***> data;

***// добавить работника***

void ***append***(const ***Worker***& person) {

data.***push\_back***(person);

}

***// обновление данных от пользователя***

***Worker*** ***stdinput***() {

***Worker*** person;

std::***string*** name, job; int year;

std::cout ***<<*** "Введите имя работника: ";

std::cin.***ignore***();

std::***getline***(std::cin, name);

std::cout ***<<*** "Введите должность работника: ";

std::***getline***(std::cin, job);

std::cout ***<<*** "Введите год принятия на работу: ";

std::cin ***>>*** year;

if (std::cin.***fail***()) {

throw ***TypeError***();

}

person.***setName***(name);

person.***setJob***(job);

person.***setYear***(year);

return person;

}

***// считать информацию из файла***

void ***filein***() {

std::***ifstream*** ***file***("data.txt");

if (!file.***is\_open***()) {

std::cerr ***<<*** "\x1b[31mНе удалось открыть файл!" ***<<*** "\x1b[0m" ***<<*** "\n";

}

std::***string*** line;

while (std::***getline***(file, line)) {

std::***istringstream*** ***iss***(line);

std::***string*** word;

std::***vector***<std::***string***> words;

while (std::***getline***(iss, word, ' ')) {

if (!word.***empty***()) {

words.***push\_back***(word);

}

}

***Worker*** person;

person.***setName***(words***[***0***]***);

person.***setJob***(words***[***1***]***);

person.***setYear***(std::***stoi***(words***[***2***]***));

***append***(person);

}

}

***// обновить информацию***

int ***update***(int& num) {

if (num < 1 || num > data.***size***()) {std::cout ***<<*** "неверный ввод!"; return 1;}

num--;

try {

data***[***num***]*** ***=*** ***stdinput***();

} catch (***TypeError***& e) {std::cout ***<<*** e.***what***();}

return 0;

}

***// удалить работника***

int ***remove***(int& num) {

if (num < 1 || num > data.***size***()) {std::cout ***<<*** "неверный ввод!"; return 1;}

num--;

data.***erase***(data.***begin***() ***+*** num);

return 0;

}

***// сортировать***

void ***sort***() {

std::***sort***(data.***begin***(), data.***end***());

}

***// поиск по имени***

void ***search\_withname***(std::***string***& name) const {

bool flag = 0;

for (int i = 0; i < data.***size***(); i++) {

if (name ***==*** data***[***i***]***.***getName***()) {

data***[***i***]***.***printInfo***();

flag = 1;

}

}

if (flag == 0) {std::cout ***<<*** "\x1b[31mнет результатов!" ***<<*** "\x1b[0m" ***<<*** "\n";}

}

***// поиск по году***

void ***search\_withyear***(int& in) const {

bool flag = 0;

for (int i = 0; i < data.***size***(); i++) {

if (in <= data***[***i***]***.***getYear***()) {

data***[***i***]***.***printInfo***();

flag = 1;

}

}

if (flag == 0) {std::cout ***<<*** "\x1b[31mнет результатов!" ***<<*** "\x1b[0m" ***<<*** "\n";}

}

***// поиск по должности***

void ***search\_withjob***(std::***string***& job) const {

bool flag = 0;

for (int i = 0; i < data.***size***(); i++) {

if (job ***==*** data***[***i***]***.***getJob***()) {

data***[***i***]***.***printInfo***();

flag = 1;

}

}

if (flag == 0) {std::cout ***<<*** "\x1b[31mнет результатов!" ***<<*** "\x1b[0m" ***<<*** "\n";}

}

***// вывести информацию***

void ***printInfo***() const {

if (data.***empty***()) {std::cout ***<<*** "\x1b[31mрабочих нет!" ***<<*** "\x1b[0m" ***<<*** "\n";}

for (int i = 0; i < data.***size***(); i++) {

data***[***i***]***.***printInfo***();

}

}

};

Результат:

