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 данных**

Отчет о лабораторной работе №3

По дисциплине «Объектно-ориентированное программирование»

ЯГТУ 09.03.04 – 006 ЛР

Работу выполнил

студент группы

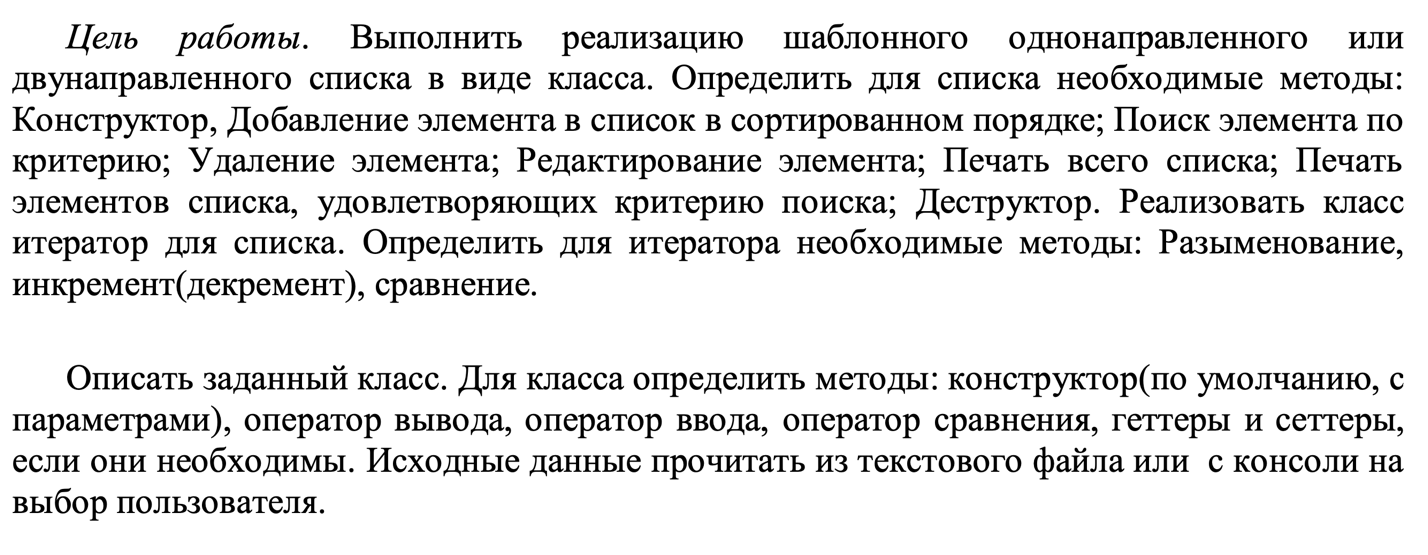
ЦПИ-11

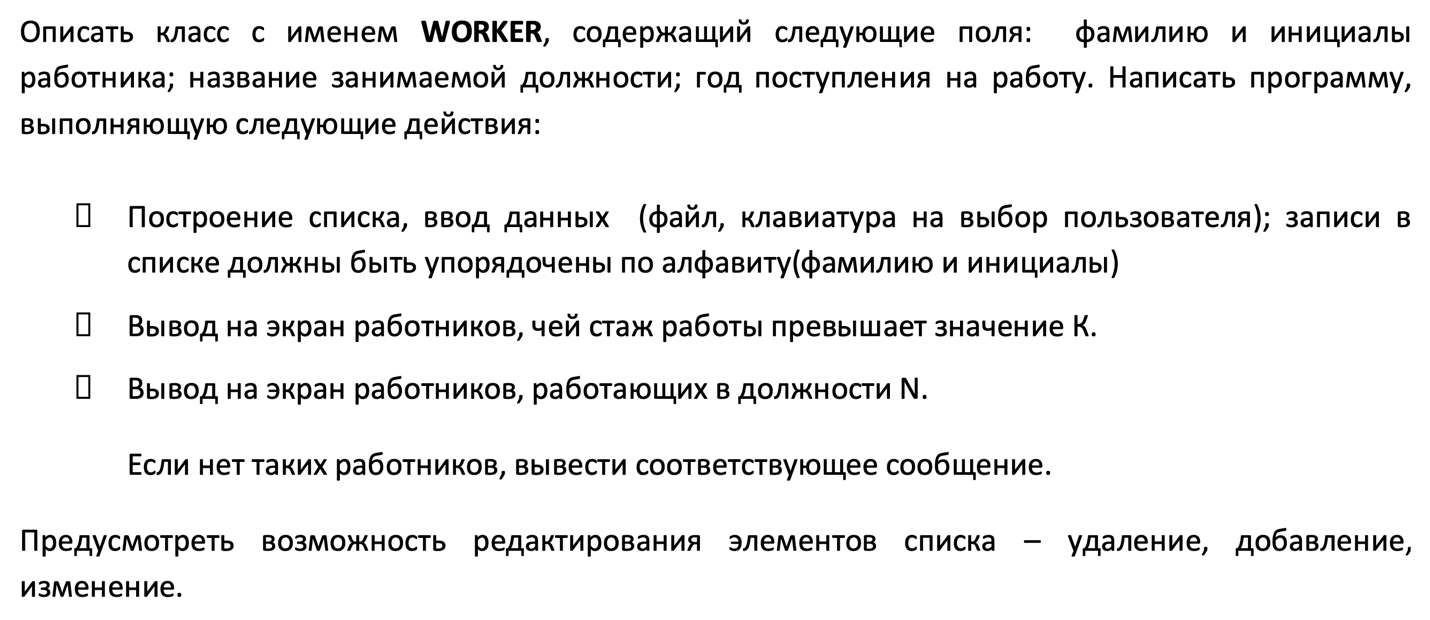
Ушаков П.А.

«10» мая 2024

Ярославль 2024

Задание:





Код:

#pragma once

#include <iostream>

struct ***value\_t*** {

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

int\* date;

};

class ***Worker*** {

public:

***value\_t*** value;

***Worker***\* next;

***Worker***(const std::***string***& name,const std::***string***& job,const int date);

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

bool ***operator>=***(const ***Worker***& other) const;

***Worker*** ***operator=***(***Worker***& other);

void ***print***() const;

void ***modify***();

};

template<typename ***T***>

class ***LinkedList*** {

***T***\* head;

***T***\* tail;

friend class ***Iterator***;

public:

***LinkedList***() {head = tail = nullptr;}

class ***Iterator*** {

***T***\* ptr;

public:

***Iterator***(***T***\* p) {ptr = p;}

***T***& ***operator\****() const {return \*ptr;}

***T***\* ***operator->***() const {return ptr;}

bool ***operator!=***(const ***Iterator***& other) const {return ptr != other.ptr;}

bool ***operator==***(const ***Iterator***& other) const {return ptr == other.ptr;}

***Iterator***& ***operator++***() {

ptr = ptr->next;

return \*this;

}

***Iterator*** ***operator++***(int) {

***Iterator*** tmp = \*this;

ptr = ptr->next;

return tmp;

}

};

***Iterator*** ***begin***() const { return ***Iterator***(head); }

***Iterator*** ***end***() const { return ***Iterator***(nullptr); }

***//void sort() {}***

void ***print***() const {for (auto i : \*this) {i.***print***();}}

***Iterator*** ***push***(***T***& node) {

***T***\* ptr = head;

***T***\* prev\_ptr = head;

if (ptr == nullptr) {

head = tail = &node;

return ***Iterator***(head);

}

while (node >= \*ptr) {

prev\_ptr = ptr;

ptr = ptr->next;

if (ptr == nullptr) {break;}

}

if (ptr == head) {node.next = ptr; head = &node; return ***Iterator***(&node);}

prev\_ptr->next = &node;

node.next = ptr;

if (ptr == nullptr) {

tail = &node;

}

return ***Iterator***(&node);

}

***Iterator*** ***popback***() {

***T***\* tmp = head;

while(tmp->next != tail) {

tmp = tmp->next;

}

tmp->next = nullptr;

***//delete tail;***

tail = tmp;

return tmp;

}

***Iterator*** ***popfront***() {

***T***\* tmp = head;

head = head->next;

delete tmp;

return ***Iterator***(head);

}

int ***del***(const int& index) {

int count = 0;

if (index < 0) {return 1;}

for (auto& i : \*this) {

if (count + 1 == index) {

***T***\* tmp = i.next;

i.next = i.next->next;

delete tmp;

return 0;

}

if (count < index) {count++;}

}

return 1;

}

int ***mod***(const int& index) {

int count = 0;

if (index < 0) {return 1;}

for (auto& i : \*this) {

if (count == index) {

i.***modify***();

return 0;

}

if (count < index) {count++;}

}

return 1;

}

};

Результат:

|  |  |
| --- | --- |
|  |  |
|  |  |