Министерство науки и высшего образования РФ

Федеральное бюджетное государственное образовательное учреждение высшего профессионального образования

Пермский национальный исследовательский политехнический университет (ПНИПУ)

Факультет: Электротехнический (ЭТФ)

Направление: 09.03.01 «Информатика и вычислительная техника (Автоматизированные системы обработки информации и управления)»

Кафедра: «Информационные технологии и автоматизированных систем» (ИТАС)

Информатика

Лабораторная работа №11

Информационные динамические структуры

Студент: Балтаев Э. А.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

(дата, подпись)

Группа: АСУ-20-1бзу

Работу проверил:

доцент кафедры ИТАС

Полякова О. А.

Оценка: \_\_\_\_\_\_\_\_\_\_\_

Подпись:\_\_\_\_\_\_\_\_\_\_\_

Пермь 2021г.

Постановка задачи:

Написать программу, в которой создаются динамические структуры и выполнить их обработку в соответствии со своим вариантом.

Задание варианта:

Сформировать двунаправленный список. Удалить из него первый элемент, добавить элемент в конец списка.

Решение задачи:

1) Листинг программы

1.1) Директивы

#define \_CRT\_SECURE\_NO\_WARNINGS

#include <iostream>

using namespace std;

1.1) Структура и переменные

struct point {

int data;

point\* next;

point\* prev;

};

FILE\* fp;

1.1) Создание элемента

point\* make\_point(int data) {

point\* p = new point;

p->data = data;

p->next = p->next = 0;

return p;

}

1.1) Создание списка

point\* make\_list() {

int n;

cout << "Количество? "; cin >> n;

if (n == 0)

return 0;

point\* beg = make\_point(rand() % 100 - 50);

point\* curr = beg;

for (int i = 1; i < n; i++) {

point\* p = make\_point(rand() % 100 - 50);

curr->next = p;

p->prev = curr;

curr = curr->next;

}

return beg;

}

1.1) Вывод списка

void print\_list(point\* beg, string text = "Список: ") {

if (beg == 0)

cout << "Пустой список";

else

cout << text;

point\* curr = beg;

while (curr != 0) {

cout << curr->data << " ";

curr = curr->next;

}

cout << endl;

}

1.1) Добавление в конец

point\* insert\_back(point\* beg) {

int val;

cout << "Значение? "; cin >> val;

point\* p = make\_point(val);

if (beg == 0)

return p;

point\* curr = beg;

while (curr->next != 0)

curr = curr->next;

curr->next = p;

p->prev = curr;

return beg;

}

1.1) Удаление первого

point\* delete\_first(point\* beg) {

if (beg == 0)

return 0;

if (beg->next == 0)

return 0;

point\* curr = beg;

beg = beg->next;

beg->prev = 0;

delete curr;

return beg;

}

1.1) Очищение списка

point\* clear\_list(point\* beg) {

while (beg != 0) {

point\* curr = beg;

beg = beg->next;

delete curr;

}

cout << "Успешное очищение списка" << endl;

return 0;

}

1.1) Сохранение в файл

int list\_to\_file(point\* beg) {

if ((fp = fopen("list.dat", "wb")) == 0)

return 1;

point\* curr = beg;

while (curr != 0) {

fwrite(&curr->data, sizeof(int), 1, fp);

curr = curr->next;

}

fclose(fp);

cout << "Успешная запись в файл" << endl;

return 0;

}

1.1) Чтение с файла

point\* list\_from\_file() {

if ((fp = fopen("list.dat", "rb")) == 0)

return 0;

int temp;

point\* beg = 0;

point\* curr = 0;

do {

fread(&temp, sizeof(int), 1, fp); // читаем

if (feof(fp))

break;

if (curr == 0) {

curr = make\_point(temp);

beg = curr;

}

else {

point\* p = make\_point(temp);

curr->next = p;

p->prev = curr;

curr = curr->next;

}

} while (true);

fclose(fp);

cout << "Успешное чтение из файла" << endl;

return beg;

}

1.1) Главная функция

int main()

{

system("chcp 1251");

srand((unsigned)time(0));

point\* beg = make\_list();

print\_list(beg, "Исходный список: ");

beg = insert\_back(beg);

print\_list(beg, "После добавления в конец: ");

beg = delete\_first(beg);

print\_list(beg, "После удаления первого: ");

int res = list\_to\_file(beg);

beg = clear\_list(beg);

print\_list(beg, "Запись в файл и уничтожение: ");

beg = list\_from\_file();

print\_list(beg, "Восстановление из файла: ");

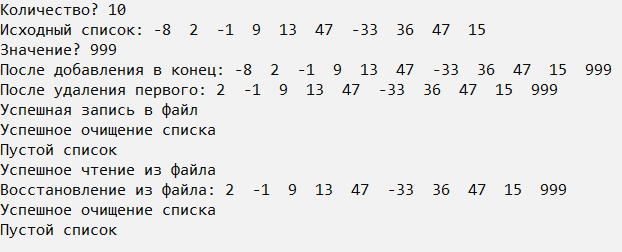
beg = clear\_list(beg);

print\_list(beg, "Уничтожение списка: ");

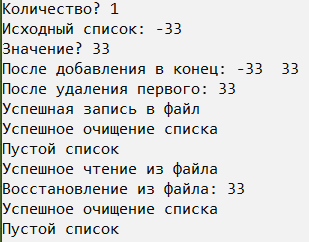
}

2) Результаты

2.1) Список из 10 элементов



2.2) Список из 1 элемента



2.3) Пустой список

