**Министерство образования и науки Российской Федерации**

**федеральное государственное бюджетное образовательное учреждение высшего профессионального образования**

**«Российский экономический университет**

**имени Г.В. Плеханова»**

**Кафедра Автоматизированных систем обработки**

**информации и управления**

## Лабораторная работа №3

**по дисциплине «Программирование (2)»**

Вариант N 3

|  |  |  |  |
| --- | --- | --- | --- |
| Исполнители  (Ф.И.О. студентов ) | Группа | Оценки по БРС  в баллах | Защита в срок |
| Петрин Михаил Сергеевич | ДКТ-151б |  |  |

Руководитель: Сорока Р.И.

Дата защиты «\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_201\_ г.

**Москва 2016**

# Оглавление

1. Общая постановка задачи лабораторной работы…………………………….3
2. Исходный код……………………………………………………………………………………..4
3. Пример работы программы……………………………………………………………….5
4. Список литературы……………………………………………………………………………..6

# Общая постановка задачи лабораторной работы.

Создать программу, которая должна состоять из двух частей, реализующих работу со списками и работу с деревьями.

**Работа со списками:**

1. Создание пустого списка.
2. Подсчет количества элементов списка.
3. Вывод на экран содержимого списка .
4. Вставка элемента в начало или в конец списка (оба варианта).
5. Исключение элемента из начала или конца списка (оба варианта).
6. Уничтожение списка с освобождением памяти.
7. Исключить элемент перед i-ым по номеру элементом.
8. Исключить i-ый+1 по номеру элемент.
9. Поменять местами элемент с заданным ключом и следующий.

**Работа с деревьями:**

1. Создание бинарного дерева в диалоге с пользователем.
2. Способ формирования дерева: с учетом значений ключа.
3. Обход бинарного дерева каждым из трех способов с выдачей на экран содержимого информационных полей.
4. Удаление заданного узла из дерева (без поддерева).
5. Удаление дерева с освобождением памяти
6. Вывод дерева на экран
7. Определить количество листьев на заданном уровне дерева.
8. Определить на каждом уровне дерева.

# Исходный код

# #include <iostream>

# #include <vector>

# #include <algorithm>

# #include <string>

# #include <cstdio>

# #include <clocale>

# #include <deque>

# #include <cmath>

# using namespace std;

# int bin(int a) {

# vector<int> v;

# while (a>0) {

# v.push\_back(a%2);

# a/=2;

# }

# v.erase(v.begin()+v.size()-1);

# a=0;

# for (int i=0; i<v.size(); ++i) {

# if (v[i]) {a+=pow(2,i);}

# }

# return a;

# }

# class tree{

# private:

# vector<int> v;

# vector<bool> used;

# public:

# int n;

# tree (){

# cout<<"Введите кол-во уровней\n";

# cin>>n;

# v.resize(pow(2,n)+1);

# used.resize(pow(2,n)+1,false);

# cout<<"Введите корень дерева\n";

# cin>>v[1];

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

# cout<<"Введите все элеметны уровня"<<" "<<i<<"\n";

# int k=0;

# for (int j=0; j<pow(2,i); ++j){

# cin>>v[pow(2,i)+k];

# k++;

# }

# }

# }

# void show (){

# for (int i=1; i<v.size()-1; ++i)

# {

# cout<<v[i]<<" ";

# if (bin(i+1)==0) cout<<endl;

# }

# }

# void dfs (int a){

# if (v[a]!=-1) {

# used[a]=1;

# cout<<v[a]<<" ";

# int k=(int)sqrt(a);

# if (k<=n-2)

# for (int p=pow(2,k+1)+(2\*(bin(a))); p<=pow(2,k+1)+(2\*bin(a))+1; ++p)

# if ((v[p]!=-1)&&(!used[p]))

# dfs(p);

# }

# }

# void dfspref ()

# {

# used.resize(pow(10,n)+1,false);

# cout<<v[1]<<" ";

# dfs(2);

# dfs(3);

# cout<<endl;

# }

# void dfsinf () {

# used.resize(pow(10,n)+1,false);

# dfs(2);

# cout<<v[1]<<" ";

# dfs(3);

# cout<<endl;

# }

# void dfspost () {

# used.resize(pow(10,n)+1,false);

# dfs(2);

# dfs(3);

# cout<<v[1]<<" ";

# cout<<endl;

# }

# void del (int a) {

# v[a]=-1;

# int k=(int)sqrt(a);

# if (k<=n-2)

# for (int p=pow(2,k+1)+(2\*(bin(a))); p<=pow(2,k+1)+(2\*bin(a))+1; ++p)

# if ((v[p]!=-1))

# del(p);

# }

# int count (int a) {

# --a;

# int sum=0;

# for (int i=pow(2,a); i<pow(2,a)\*2;++i) {

# if(v[i]!=-1) {

# ++sum;

# }

# }

# return sum;

# }

# void all\_count () {

# for (int i=2; i<=n+1; ++i) {

# cout<<"Количество элементов на данном уровне "<<tree::count(i)<<endl;

# }}};

# void menu (){

# cout<<"1:Создание бинарное дерево"<<endl;

# cout<<"2:Обход бинарного дерева 1 метод."<<endl;

# cout<<"3:Обход бинарного дерева 2 метод."<<endl;

# cout<<"4:Обход бинарного дерева 3 метод."<<endl;

# cout<<"5:Удаление заданного узла из дерева "<<endl;

# cout<<"6:Удаление дерева с освобождением памяти"<<endl;

# cout<<"7:Вывод дерева на экран"<<endl;

# cout<<"8:Определить количество листьев на заданном уровне дерева"<<endl;

# cout<<"9:Определить на каждом уровне дерева."<<endl;

# cout<<"10:Выход"<<endl;

# }

# void show (deque<char> s){

# for (int i=0; i<s.size(); ++i)

# cout<<s[i];

# cout<<endl;

# }

# void menu1 (){

# cout<<"1:Создание пустого списка"<<endl;

# cout<<"2:Подсчет количества элементов списка"<<endl;

# cout<<"3:Вывод на экран содержимого списка"<<endl;

# cout<<"4:Вставка элемента в начало списка"<<endl;

# cout<<"5:Вставка элемента в конец списка"<<endl;

# cout<<"6:Исключение элемента из начала списка"<<endl;

# cout<<"7:Исключение элемента из конца списка"<<endl;

# cout<<"8:Уничтожение списка с освобождением памяти."<<endl;

# cout<<"9:Исключить элемент перед i-ым по номеру элементом."<<endl;

# cout<<"10:Исключить i-ый+1 по номеру элемент."<<endl;

# cout<<"11:Поменять местами i i+1"<<endl;

# cout<<"12:Выйти"<<endl;

# }

# int strock (){

# cout<<"Работа со списками\n";

# cout<<"Введите список"<<endl;

# string ss;

# cin>>ss;

# deque<char>s (ss.size());

# char c;

# for (int i=0; i<ss.size(); ++i)

# s[i]=ss[i];

# int a;

# int b;

# show(s);

# while (true) {

# menu1();

# cin>>a;

# switch (a) {

# case 1: cout<<"Список создан"<<endl; continue;

# case 2: cout<<"Всего элементом в списке"<<" "<<s.size()<<endl; continue;

# case 3: show(s); continue;

# case 4: cout<<"Введите элемент\n"; cin>>c; s.push\_front(c); cout<<"Элемент вставлен-";show(s); continue;

# case 5: cout<<"Введите элемент\n"; cin>>c; s.push\_back(c); cout<<"Элемент вставлен-";show(s); continue;

# case 6: s.erase(s.begin()); cout<<"Элемент удален";show(s); continue;

# case 7: s.erase(s.begin()+s.size()-1); cout<<"Элемент удален";show(s); continue;

# case 8: s.~deque(); continue;

# case 9: cout<<"Введите индекс"; cin>>b; s.erase(s.begin()+b-1); show(s); continue;

# case 10: cout<<"Введите индекс"; cin>>b; s.erase(s.begin()+b); show(s); continue;

# case 11: cout<<"Введите символ"; cin>>c; swap(s[c],s[c+1]); continue;

# case 12: return 0;

# default: cout<<"Ошибка ввода"<<endl;

# }}}

# int der (){

# setlocale(LC\_ALL,"rus");

# tree t;

# while(true) {

# menu();

# int a;

# cin>>a;

# switch(a) {

# case 1 : cout<<"Дерево создано"<<endl; continue;

# case 2 : t.dfsinf();continue;

# case 3 : t.dfspref();continue;

# case 4 : t.dfspost();continue;

# case 5 : int b; cin>>b; t.del(b); continue;

# case 6 : t.del(0); continue;

# case 7 : t.show(); continue;

# case 8 : int c; cin>>c; cout<<t.count(c)<<endl; continue;

# case 9 : t.all\_count(); continue;

# case 10 : return 0;

# default: cout<<"Ошибка ввода"<<endl;

# }}}

# int main(){

# setlocale(LC\_ALL,"rus");

# while (true){

# cout<<"1:Работа с деревьями"<<endl;

# cout<<"2:Работа с строками"<<endl;

# cout<<"3: Выход"<<endl;

# int a;

# cin>>a;

# switch (a){

# case 1: der();continue;

# case 2: strock();continue;

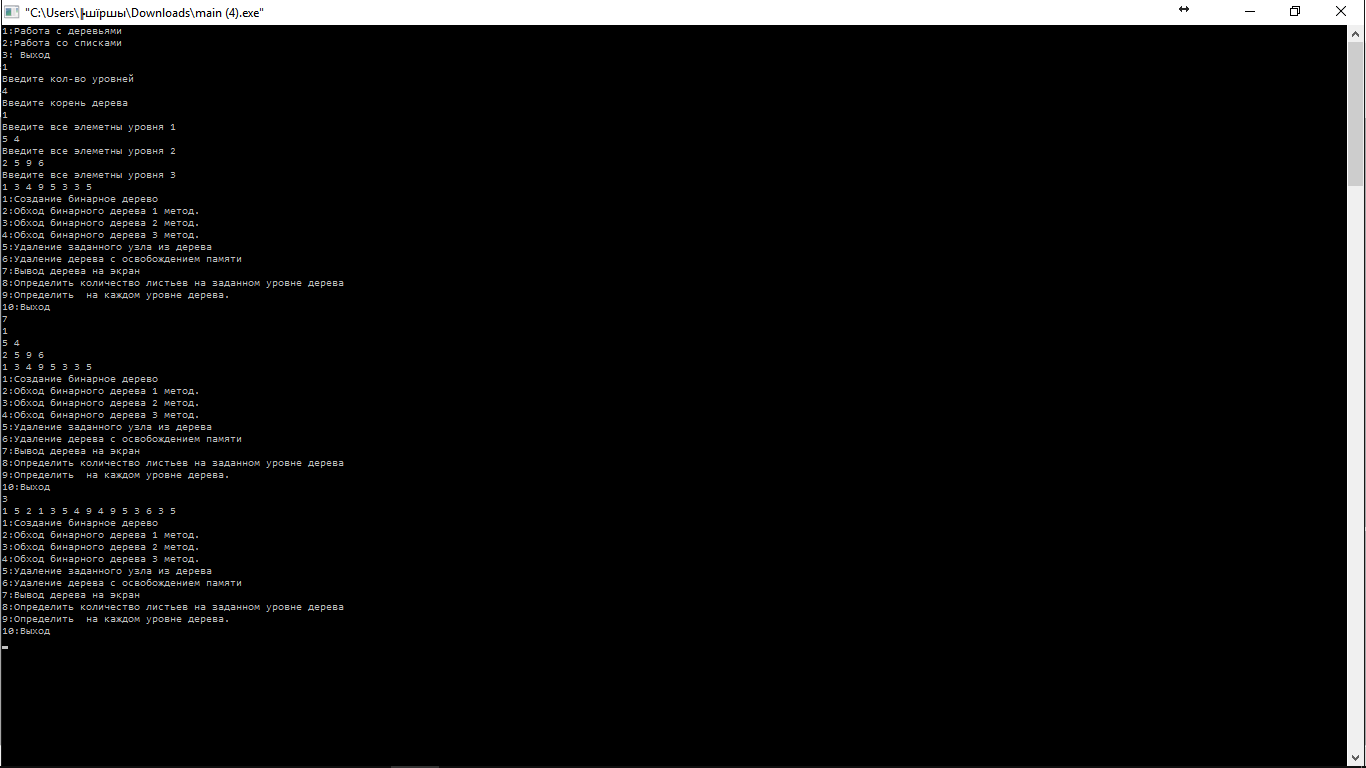
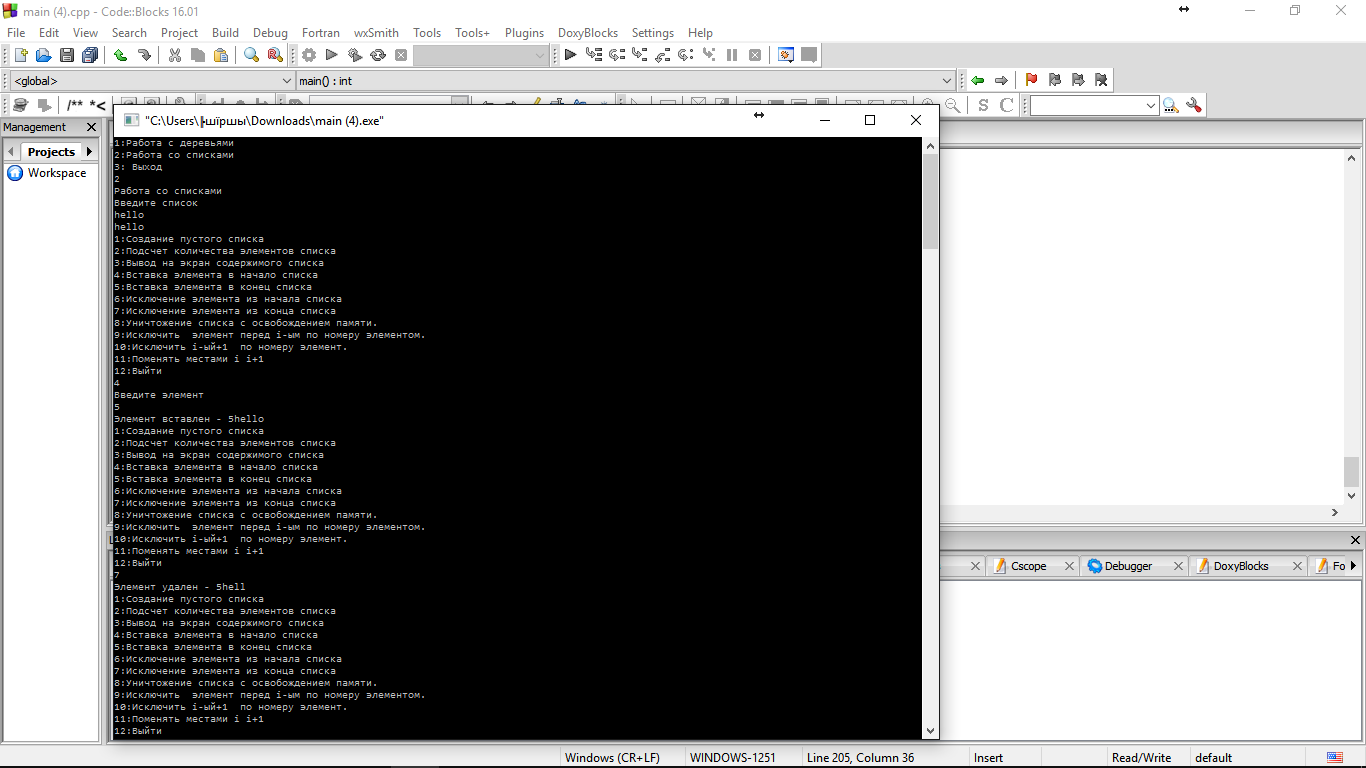
# case 3: return 0;

# }

# }return 0;

# }

# Пример работы программы



# Список литературы:

1. *Бьёрн Страуструп.* Язык программирования C++ = The C++ Programming Language / Пер. с англ. — 3-е изд. — СПб.;