Федеральное государственное бюджетное образовательное учреждение высшего образования «Российский экономический университет имени Г.В. Плеханова»   
**Московский приборостроительный техникум**

**Отчет**   
о выполнении практической работы № 8

Дисциплина: ОП.04 «Основы алгоритмизации и программирования»

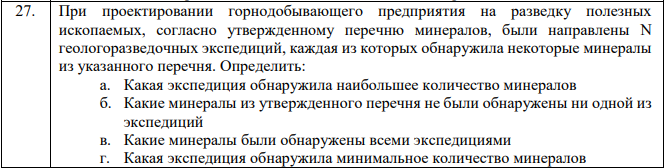
Тема: Работа с данными типа множество

Специальность: 09.02.07 Информационные системы и программирование

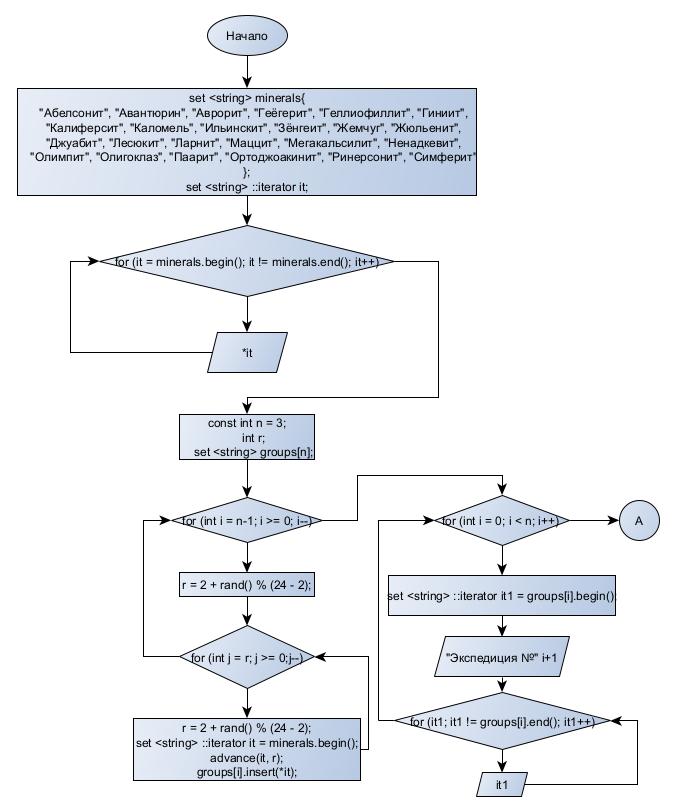
Квалификация: Специалист по информационным системам

Выполнил: Dusky Fox  
Студент группы:   
Вариант: № 27  
Дата: «11» декабря 2021 г.

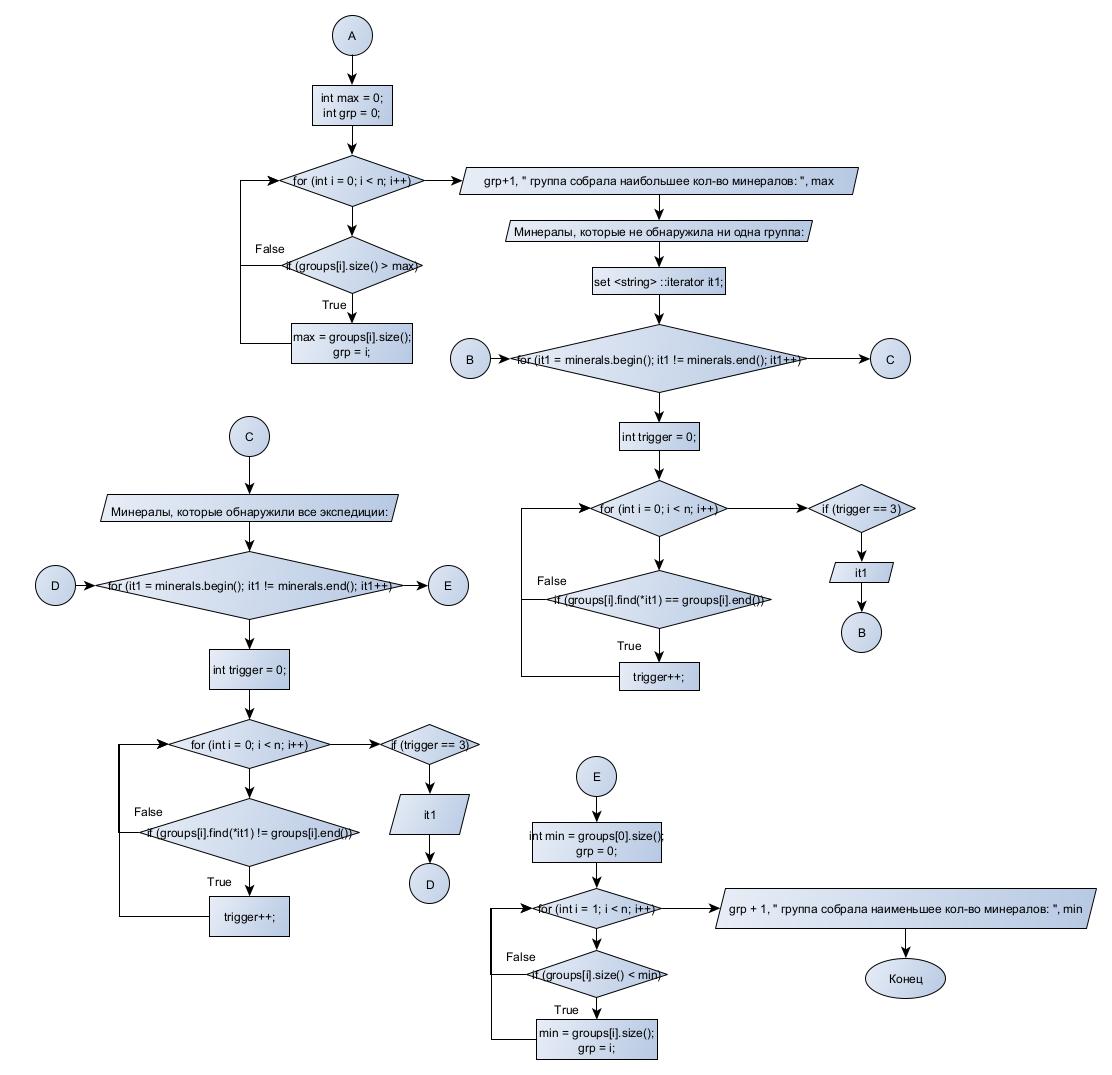
Проверил:   
Преподаватель: Клопов Д.А.  
Дата: «\_\_\_» \_\_\_\_\_\_\_\_\_ 202\_\_ г.   
Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



1Задание



2Блок-схема, первая часть



3Блок-схема вторая часть

#include <string>

#include <iostream>

#include <Windows.h>

#include <set>

using namespace std;

set <string> minerals{

    "Абелсонит", "Авантюрин", "Аврорит", "Геёгерит", "Геллиофиллит", "Гиниит",

    "Калиферсит", "Каломель", "Ильинскит", "Зёнгеит", "Жемчуг", "Жюльенит",

    "Джуабит", "Лесюкит", "Ларнит", "Маццит", "Мегакальсилит", "Ненадкевит",

    "Олимпит", "Олигоклаз", "Паарит", "Ортоджоакинит", "Ринерсонит", "Симферит"

};

int main()

{

    SetConsoleCP(1251);

    SetConsoleOutputCP(1251);

    set <string> ::iterator it;

    cout << "Все минералы: \n";

    for (it = minerals.begin(); it != minerals.end(); it++) {

        cout << \*it << ((\*it != \*minerals.rbegin()) ? ", " : "");

    }

    cout << endl << "\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_" << endl;

    const int n = 3;

    int r;

    set <string> groups[n];

    for (int i = n-1; i >= 0; i--) {

        r = 2 + rand() % (24 - 2);

        for (int j = r; j >= 0;j--) {

            r = 2 + rand() % (24 - 2);

            set <string> ::iterator it = minerals.begin();

            advance(it, r);

            groups[i].insert(\*it);

        }

    }

    for (int i = 0; i < n; i++) {

        set <string> ::iterator it1 = groups[i].begin();

        cout << "Экспедиция №" << i+1 << endl;

        for (it1; it1 != groups[i].end(); it1++)

                cout << \*it1 << ((\*it1 != \*groups[i].rbegin()) ? ", " : "");

        cout << "\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

    }

    int max = 0;

    int grp = 0;

    for (int i = 0; i < n; i++) {

        if (groups[i].size() > max) {

            max = groups[i].size();

            grp = i;

        }

    }

    cout << endl << grp+1 << " группа собрала наибольшее кол-во минералов: " << max;

    cout << "\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

    cout << "\nМинералы, которые не обнаружила ни одна группа:\n";

    set <string> ::iterator it1;

    for (it1 = minerals.begin(); it1 != minerals.end(); it1++) {

        int trigger = 0;

        for (int i = 0; i < n; i++) {

            if (groups[i].find(\*it1) == groups[i].end())

                trigger++;

        }

        if (trigger == 3) {

            cout << \*it1 << ", ";

        }

    }

    cout << "\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

    cout << "\nМинералы, которые обнаружили все экспедиции:\n";

    for (it1 = minerals.begin(); it1 != minerals.end(); it1++) {

        int trigger = 0;

        for (int i = 0; i < n; i++) {

            if (groups[i].find(\*it1) != groups[i].end())

                trigger++;

        }

        if (trigger == 3) {

            cout << \*it1 << ", ";

        }

    }

    cout << "\n\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\n";

    int min = groups[0].size();

    grp = 0;

    for (int i = 1; i < n; i++) {

        if (groups[i].size() < min) {

            min = groups[i].size();

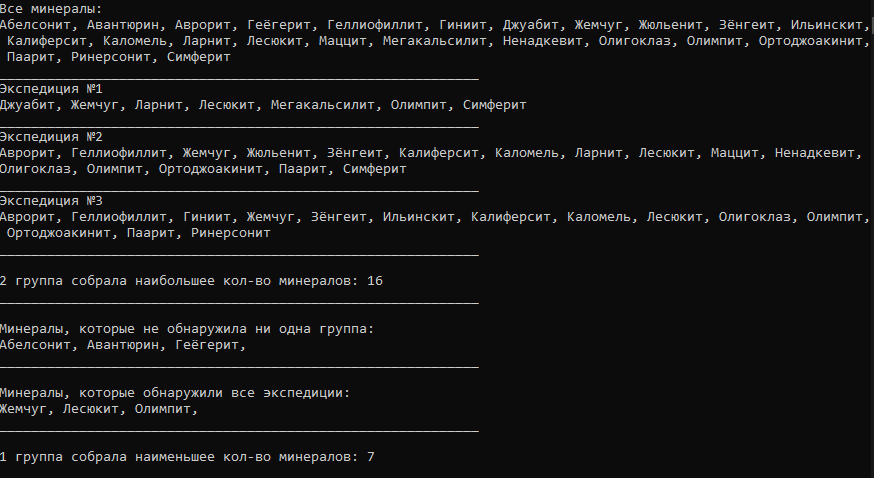
            grp = i;

        }

    }

    cout << endl << grp + 1 << " группа собрала наименьшее кол-во минералов: " << min << endl;

}



4Вывод программы в консоль