

Лабораторна робота 13. АЛГОРИТМИ ПЕРЕМІЩЕННЯ ТА ПОШУКУ

Тема: STL. Алгоритми переміщення та пошуку.

Мета: на практиці порівняти STL-алгоритми, що не модифікують послідовність.

1. Завдання до роботи

Загальне завдання. Поширити попередню лабораторну роботу, додаючи такі можливості діалогового меню:

- виведення всіх елементів масиву за допомогою *STL-функції for_each*;
- визначення кількості елементів за заданим критерієм;
- пошук елемента за заданим критерієм.

2.1. Опис класів

Базовий клас: `CProgram`.

Клас-спадкоємець: `CMalware`.

2.2. Опис змінних

`int` `timeOfWork` – час роботи програми (змінна класу `CProgram`).
`int` `size` – розмір програми (змінна класу `CProgram`).
`int` `amountOfLines` – кількість рядків коду програми (змінна класу `CProgram`).
`int` `index` – номер програми (змінна класу `CProgram`).
`bool` `useInternet` – потребує програма Інтернет чи ні (змінна класу `CProgram`).
`string` `name` – назва програми (змінна класу `CProgram`).
`string` `type` – тип зловмисного ПО (змінна класу `CMalware`).

2.3. Опис методів

`virtual string` `getInfo() const` – виведення даних елемента у консоль (метод класу `CProgram`).
`virtual stringstream` `getStr() const` – отримання строки з даними елемента (метод класу `CProgram`).
`int` `getID() const` – отримання індекса елемента (метод класу `CProgram`).
`bool` `elementOutput(int, string)` – виведення елемента за обраним критерієм (метод класу `CProgram`).

`int countElement(int, string)` – виведення кількості елементів за обраним критерієм (метод класу `CProgram`).

`CProgram()` – конструктор класу за замовчуванням (метод класу `CProgram`).

`CProgram(bool, int, int, int, int, string)` – конструктор класу з параметрами (метод класу `CProgram`).

`CProgram(const CProgram&)` – конструктор копіювання (метод класу `CProgram`).

`virtual ~CProgram()` – деструктор класу (метод класу `CProgram`).

`friend ostream& operator<< (ostream&, const CProgram&)` – перевантаження оператора `<<` (метод класу `CProgram`).

`virtual bool operator==(const int) const` – перевантаження оператора `==` (метод класу `CProgram`).

3. Текст програми

main.cpp

```
#include "malware.h"

CProgram* newProgram(int);
void VectorMenu();
void ListMenu();
void MapMenu();
void SetMenu();

int main()
{
    setlocale(LC_ALL, "Rus");
    int choise = 0;
    bool stop = 1;

    while (stop)
    {
        cout << "Выберите STL контейнер:" << endl;
        cout << "1. Vector" << endl;
        cout << "2. List" << endl;
        cout << "3. Map" << endl;
        cout << "4. Set" << endl;
        cout << "5. Выход" << endl;
        cout << "======" << endl;
        cout << "Ваш выбор: ";
        cin >> choise;

        switch (choise)
        {
            case 1:
                VectorMenu();
                break;

            case 2:
                ListMenu();
                break;

            case 3:
                MapMenu();
                break;

            case 4:
                SetMenu();
                break;

            case 5:
                stop = 0;
        }
    }
}
```

```

        break;

    default:
        cout << "Ошибка. Неверная команда. Повторите попытку." << endl;
    }
}

if (_CrtDumpMemoryLeaks())
    cout << endl << "Есть утечка памяти." << endl;
else
    cout << endl << "Утечка памяти отсутствует." << endl;

return 0;
}

CProgram* newProgram(int value)
{
    if (value % 2 == 0)
    {
        CProgram* temp = new CMalware(1, 5231, 505, 101, 56234, "KeySaver", "Keylogger");
        return temp;
    }
    else
    {
        CProgram* temp = new CProgram(0, 645, 634, 6745, 45678, "Photoshop");
        return temp;
    }
}

void VectorMenu()
{
    vector <unique_ptr<CProgram>> vector;
    std::vector<unique_ptr<CProgram>>::iterator it;
    stringstream temp;
    string data;
    bool stop = 1, findEl = 0;
    int choise = 0, choise2 = 0, choise3 = 0;
    int value = 0, number = 0, result = 0, sum = 0;

    for (size_t i = 0; i < 4; i++)
    {
        if (i == 0)
            vector.emplace_back(new CProgram());
        else if (i == 1)
            vector.emplace_back(new CMalware(1, 8800, 555, 35, 35634, "BestMalware",
"Exploit"));
        else if (i == 2)
            vector.emplace_back(new CProgram(0, 423, 523, 654, 53453, "Calculator"));
        else if (i == 3)
            vector.emplace_back(new CMalware(0, 345, 789, 423, 67456, "MoneyStealer",
"Rootkit"));
    }

    while (stop != 0)
    {
        if (vector.size() == 0)
        {
            cout << "Вектор пуст. Что вы хотите сделать?" << endl;
            cout << "1) Добавить элемент" << endl;
            cout << "2) Завершение работы" << endl;
            cout << "===== " << endl;
            cout << "Ваш выбор: ";
            cin >> choise;
            cout << endl;

            switch (choise)
            {
                case 1:
                    cout << "Выберите программу, которую хотите добавить:" << endl;

```

```

        cout << "1. Элемент класса CProgram" << endl;
        cout << "2. Элемент класса CMalware" << endl;
        cout << "======" << endl;
        cout << "Ваш выбор: ";
        cin >> value;

        try
        {
            vector.at(value);

            if (value == 1 || value == 2)
            {
                vector.emplace_back(newProgram(value));
                cout << "Элемент добавлен." << endl;
            }
            else
                cout << "Ошибка. Неверный номер." << endl;
        }
        catch (const std::exception& ex)
        {
            cout << ex.what() << endl;
        }

        break;

    case 2:
        cout << "Завершение работы." << endl;
        stop = 0;
        break;

    default:
        cout << "Неверный номер элемента. Повторите попытку." << endl;
        break;
    }
}
else
{
    cout << endl;
    cout << "1) Вывод на экран" << endl;
    cout << "2) Удаление элемента" << endl;
    cout << "3) Добавление элементов" << endl;
    cout << "4) Завершение работы" << endl;
    cout << "======" << endl;
    cout << "Ваш выбор: ";
    cin >> choise;
    cout << endl;
}

switch (choise)
{
    case 1:
        cout << "Выберите команду:" << endl;
        cout << "1) Вывести весь список на экран" << endl;
        cout << "2) Вывести программу по ID" << endl;
        cout << "3) Вывести количество элементов по критерию" << endl;
        cout << "4) Найти элемент по критерию" << endl;
        cout << "5) Вернуться к выбору действий" << endl;
        cout << "======" << endl;
        cout << "Ваш выбор: ";
        cin >> choise2;
        cout << endl;

        switch (choise2)
        {
            case 1:
                cout << setw(12) << "Название" << setw(14) << "Индекс";
                cout << setw(14) << "Время работы" << setw(8) << "Размер";
                cout << setw(18) << "Количество линий" << setw(10) << "Интернет";
                cout << setw(10) << "Тип" << endl;

```

```

        number = 1;
        for_each(vector.begin(), vector.end(), [&number](const
unique_ptr<CProgram>& program)
        {
            cout << number << ". " << *program << endl;
            number++;
        });
        number = 1;
        break;

case 2:
    cout << "Введите id элемента, которого вы хотите получить: ";
    cin >> value;
    cout << endl;

    findEl = 0, number = -1;
    for (const auto& element: vector)
    {
        if (element->getID() == value)
        {
            number++;
            findEl = 1;
            break;
        }
        else
            number++;
    }

    if (findEl)
    {
        temp = vector[number]->getStr();
        data = temp.str();
        cout << "Ваш элемент: " << endl;
        cout << data << endl << endl;
    }
    else
        cout << "Элемент с таким ID не найден." << endl;

    break;

case 3:
    cout << "Выберите критерий, по которому надо искать: " << endl;
    cout << "1) Название" << endl;
    cout << "2) Время работы" << endl;
    cout << "3) Размер" << endl;
    cout << "4) Количество строк кода" << endl;
    cout << "5) Индекс" << endl;
    cout << "6) Использует ли интернет" << endl;
    cout << "7) Вернуться назад" << endl;
    cout << "======" << endl;
    cout << "Ваш выбор: ";
    cin >> choise3;
    cout << endl;

    if (choise3 < 1 || choise3 >= 7)
    {
        cout << "Возвращение назад." << endl;
        break;
    }

    it = vector.begin();

    cout << "Введите критерий: ";
    cin.ignore();
    getline(cin, data);
    number = 0, value = 0;

    while (number < vector.size())
    {

```

```

        result = (*it)->countElement(choise3, data);
        number++;
        it++;
        sum += result;
    }
    if (sum != 0)
        cout << "Количество элементов с данным параметром: " << sum <<
endl;

    break;

case 4:
    cout << "Выберите критерий, по которому надо искать: " << endl;
    cout << "1) Название" << endl;
    cout << "2) Время работы" << endl;
    cout << "3) Размер" << endl;
    cout << "4) Количество строк кода" << endl;
    cout << "5) Индекс" << endl;
    cout << "6) Использует ли интернет" << endl;
    cout << "7) Вернуться назад" << endl;
    cout << "===== " << endl;
    cout << "Ваш выбор: ";
    cin >> choise3;
    cout << endl;

    if (choise3 < 1 || choise3 >= 7)
    {
        cout << "Возвращение назад." << endl;
        break;
    }

    it = vector.begin();
    cout << "Введите критерий: ";
    cin.ignore();
    getline(cin, data);
    number = 0, value = 0;

    while (number < vector.size())
    {
        result = (*it)->elementOutput(choise3, data);
        number++;
        it++;
    }

    break;
case 5:
    cout << "Возвращение назад." << endl;
    break;

default:
    cout << "Неверный символ. Повторите попытку." << endl;
    break;

}
break;

case 2:
    cout << "Введите ID элемента, который хотите удалить: ";
    cin >> value;
    cout << endl;

    findEl = 0, number = -1;
    for (const auto& element:vector)
    {
        if (element->getID() == value)
        {
            number++;
            findEl = 1;
            break;

```

```

        }
        else
            number++;
    }

    if (findEl)
    {
        it = vector.begin();
        advance(it, number);
        vector.erase(it);

        cout << "Удаление выполнено." << endl;
    }
    else
        cout << "Элемент не найден." << endl;

    break;

case 3:
    cout << "Выберите программу, которую хотите добавить:" << endl;
    cout << "1. Элемент класса CProgram" << endl;
    cout << "2. Элемент класса CMalware" << endl;
    cout << "===== " << endl;
    cout << "Ваш выбор: ";
    cin >> value;

    try
    {
        vector.at(value);

        if (value == 1 || value == 2)
        {
            vector.emplace_back(newProgram(value));
            cout << "Элемент добавлен." << endl;
        }
        else
            cout << "Ошибка. Неверный номер." << endl;
    }
    catch (const std::exception & ex)
    {
        cout << ex.what() << endl;
    }

    break;

case 4:
    cout << "Завершение работы." << endl << endl;
    stop = 0;
    break;

default:
    cout << "Неверный символ. Повторите попытку." << endl;
    break;
}

}

}

void ListMenu()
{
    list <unique_ptr<CProgram>> list;
    stringstream temp;
    string data;
    bool stop = 1, findEl = 0;
    int choise = 0, choise2 = 0, choise3 = 0;
    int value = 0;
    int number = 0;
    int result = 0, sum = 0;
    auto it = list.begin();

```

```

for (size_t i = 0; i < 4; i++)
{
    if (i == 0)
        list.emplace_back(new CProgram());
    else if (i == 1)
        list.emplace_back(new CMalware(1, 8800, 555, 35, 35634, "BestMalware",
"Exploit"));
    else if (i == 2)
        list.emplace_back(new CProgram(0, 423, 523, 654, 53453, "Calculator"));
    else if (i == 3)
        list.emplace_back(new CMalware(0, 345, 789, 423, 67456, "MoneyStealer",
"Rootkit"));
}

while (stop != 0)
{
    if (list.size() == 0)
    {
        cout << "Вектор пуст. Что вы хотите сделать?" << endl;
        cout << "1) Добавить элемент" << endl;
        cout << "2) Завершение работы" << endl;
        cout << "===== " << endl;
        cout << "Ваш выбор: ";
        cin >> chose;
        cout << endl;

        switch (choise)
        {
            case 1:
                cout << "Выберите программу, которую хотите добавить:" << endl;
                cout << "1. Элемент класса CProgram" << endl;
                cout << "2. Элемент класса CMalware" << endl;
                cout << "===== " << endl;
                cout << "Ваш выбор: ";
                cin >> value;

                try
                {
                    if (value == 1 || value == 2)
                    {
                        list.emplace_front(newProgram(value));
                        cout << "Элемент добавлен." << endl;
                    }
                    else
                        cout << "Ошибка. Неверный номер." << endl;
                }
                catch (const std::exception & ex)
                {
                    cout << ex.what() << endl;
                }

                break;

            case 2:
                cout << "Завершение работы." << endl;
                stop = 0;
                break;

            default:
                cout << "Неверный номер элемента. Повторите попытку." << endl;
                break;
        }
    }
    else
    {
        cout << endl;
        cout << "1) Вывод на экран" << endl;
        cout << "2) Удаление элемента" << endl;
        cout << "3) Добавление элементов" << endl;
    }
}

```



```

        cout << "4)Завершение работы" << endl;
        cout << "======" << endl;
        cout << "Ваш выбор: ";
        cin >> choise;
        cout << endl;
    }

    switch (choise)
    {
    case 1:
        cout << "Выберите команду:" << endl;
        cout << "1) Вывести весь список на экран" << endl;
        cout << "2) Вывести программу по ID" << endl;
        cout << "3) Вывести количество элементов по критерию" << endl;
        cout << "4) Найти элемент по критерию" << endl;
        cout << "5) Вернуться к выбору действий" << endl;
        cout << "======" << endl;
        cout << "Ваш выбор: ";
        cin >> choise2;
        cout << endl;

        switch (choise2)
        {
        case 1:
            cout << setw(12) << "Название" << setw(14) << "Индекс";
            cout << setw(14) << "Время работы" << setw(8) << "Размер";
            cout << setw(18) << "Количество линий" << setw(10) << "Интернет";
            cout << setw(10) << "Тип" << endl;

            number = 1;
            for_each(list.begin(), list.end(), [&number](const unique_ptr<CProgram>&
program)
                {
                    cout << number << ". " << *program << endl;
                    number++;
                });
            number = 1;
            break;

        case 2:
            cout << "Введите id элемента, которого вы хотите получить: ";
            cin >> value;
            cout << endl;

            findEl = 0, number = -1;
            for (const auto& element : list)
            {
                if (element->getID() == value)
                {
                    number++;
                    findEl = 1;
                    break;
                }
                else
                    number++;
            }

            if (findEl)
            {
                it = list.begin();
                advance(it, number);

                temp = (*it)->getStr();
                data = temp.str();

                cout << "Ваш элемент: " << endl;
                cout << data << endl << endl;
            }
            else

```

```

        cout << "Элемент с таким ID не найден." << endl;

        break;

case 3:
    cout << "Выберите критерий, по которому надо искать: " << endl;
    cout << "1) Название" << endl;
    cout << "2) Время работы" << endl;
    cout << "3) Размер" << endl;
    cout << "4) Количество строк кода" << endl;
    cout << "5) Индекс" << endl;
    cout << "6) Использует ли интернет" << endl;
    cout << "7) Вернуться назад" << endl;
    cout << "===== " << endl;
    cout << "Ваш выбор: ";
    cin >> choise3;
    cout << endl;

    if (choise3 < 1 || choise3 >= 7)
    {
        cout << "Возвращение назад." << endl;
        break;
    }

    it = list.begin();
    result = 0, sum = 0;
    cout << "Введите критерий: ";
    cin.ignore();
    getline(cin, data);
    number = 0, value = 0;

    while (number < list.size())
    {
        result = (*it)->countElement(choise3, data);
        number++;
        it++;
        sum += result;
    }
    if (sum != 0)
        cout << "Количество элементов с данным параметром: " << sum <<
endl;

    break;

case 4:
    cout << "Выберите критерий, по которому надо искать: " << endl;
    cout << "1) Название" << endl;
    cout << "2) Время работы" << endl;
    cout << "3) Размер" << endl;
    cout << "4) Количество строк кода" << endl;
    cout << "5) Индекс" << endl;
    cout << "6) Использует ли интернет" << endl;
    cout << "7) Вернуться назад" << endl;
    cout << "===== " << endl;
    cout << "Ваш выбор: ";
    cin >> choise3;
    cout << endl;

    if (choise3 < 1 || choise3 >= 7)
    {
        cout << "Возвращение назад." << endl;
        break;
    }

    it = list.begin();
    cout << "Введите критерий: ";
    cin.ignore();
    getline(cin, data);
    number = 0, value = 0;

```

```

        while (number < list.size())
        {
            result = (*it)->elementOutput(choise3, data);
            number++;
            it++;
        }

        break;
case 5:
    cout << "Возвращение назад." << endl;
    break;

default:
    cout << "Неверный символ. Повторите попытку." << endl;
    break;

    }
    break;

case 2:
    cout << "Введите ID элемента, который хотите удалить: ";
    cin >> value;
    cout << endl;

    findEl = 0, number = -1;
    for (const auto& element : list)
    {
        if (element->getID() == value)
        {
            number++;
            findEl = 1;
            break;
        }
        else
            number++;
    }

    if (findEl)
    {
        it = list.begin();
        advance(it, number);
        list.erase(it);

        cout << "Удаление выполнено." << endl;
    }
    else
        cout << "Элемент не найден." << endl;

    break;

case 3:
    cout << "Выберите программу, которую хотите добавить:" << endl;
    cout << "1. Элемент класса CProgram" << endl;
    cout << "2. Элемент класса CMalware" << endl;
    cout << "===== " << endl;
    cout << "Ваш выбор: ";
    cin >> value;

    try
    {
        if (value == 1 || value == 2)
        {
            list.emplace_front(newProgram(value));
            cout << "Элемент добавлен." << endl;
        }
        else
            cout << "Ошибка. Неверный номер." << endl;
    }

```

```

        catch (const std::exception & ex)
        {
            cout << ex.what() << endl;
        }

        break;

    case 4:
        cout << "Завершение работы." << endl << endl;
        stop = 0;
        break;

    default:
        cout << "Неверный символ. Повторите попытку." << endl;
        break;
    }
}

void MapMenu()
{
    map <int, unique_ptr<CProgram>> map;
    stringstream temp;
    string data;
    bool stop = 1, findEl = 0;
    int chose = 0, chose2 = 0, chose3 = 0;
    int value = 0;
    int i = 0;
    int number = 0, sum = 0, result = 0;
    auto it = map.begin();

    for (; i < 4; i++)
    {
        if (i == 0)
            map.emplace(i + 1, new CProgram());
        else if (i == 1)
            map.emplace(i + 1, new CMalware(1, 8800, 555, 35, 35634, "BestMalware",
"Exploit"));
        else if (i == 2)
            map.emplace(i + 1, new CProgram(0, 423, 523, 654, 53453, "Calculator"));
        else if (i == 3)
            map.emplace(i + 1, new CMalware(0, 345, 789, 423, 67456, "MoneyStealer",
"Rootkit"));
    }

    while (stop != 0)
    {
        if (map.size() == 0)
        {
            cout << "Вектор пуст. Что вы хотите сделать?" << endl;
            cout << "1) Добавить элемент" << endl;
            cout << "2) Завершение работы" << endl;
            cout << "=====" << endl;
            cout << "Ваш выбор: ";
            cin >> chose;
            cout << endl;

            switch (chose)
            {
            case 1:
                cout << "Выберите программу, которую хотите добавить:" << endl;
                cout << "1. Элемент класса CProgram" << endl;
                cout << "2. Элемент класса CMalware" << endl;
                cout << "=====" << endl;
                cout << "Ваш выбор: ";
                cin >> value;

                try
                {

```

```

        if (value == 1 || value == 2)
        {
            map.emplace(++i, newProgram(value));
            cout << "Элемент добавлен." << endl;
        }
        else
            cout << "Ошибка. Неверный номер." << endl;
    }
    catch (const std::exception & ex)
    {
        cout << ex.what() << endl;
    }

    break;

case 2:
    cout << "Завершение работы." << endl;
    stop = 0;
    break;

default:
    cout << "Неверный номер элемента. Повторите попытку." << endl;
    break;
}
}
else
{
    cout << endl;
    cout << "1) Вывод на экран" << endl;
    cout << "2) Удаление элемента" << endl;
    cout << "3) Добавление элементов" << endl;
    cout << "4) Завершение работы" << endl;
    cout << "======" << endl;
    cout << "Ваш выбор: ";
    cin >> choise;
    cout << endl;
}

switch (choise)
{
case 1:
    cout << "Выберите команду:" << endl;
    cout << "1) Вывести весь список на экран" << endl;
    cout << "2) Вывести программу по ID" << endl;
    cout << "3) Вывести количество элементов по критерию" << endl;
    cout << "4) Найти элемент по критерию" << endl;
    cout << "5) Вернуться к выбору действий" << endl;
    cout << "======" << endl;
    cout << "Ваш выбор: ";
    cin >> choise2;
    cout << endl;

    switch (choise2)
    {
case 1:
        cout << setw(12) << "Название" << setw(14) << "Индекс";
        cout << setw(14) << "Время работы" << setw(8) << "Размер";
        cout << setw(18) << "Количество линий" << setw(10) << "Интернет";
        cout << setw(10) << "Тип" << endl;

        for_each(map.begin(), map.end(), [](const std::pair<const int,
unique_ptr<CProgram>>& program)
        {
            cout << program.first << ". " << *program.second << endl;
        });

        break;

case 2:

```

```

cout << "Введите номер элемента, которого вы хотите получить: ";
cin >> value;
cout << endl;

findEl = 0;
it = map.find(value);

if (it != map.end())
{
    temp = (*it).second->getStr();
    data = temp.str();

    cout << "Ваш элемент: " << endl;
    cout << data << endl << endl;
}
else
    cout << "Элемент с таким ID не найден." << endl;

break;

```

case 3:

```

cout << "Выберите критерий, по которому надо искать: " << endl;
cout << "1) Название" << endl;
cout << "2) Время работы" << endl;
cout << "3) Размер" << endl;
cout << "4) Количество строк кода" << endl;
cout << "5) Индекс" << endl;
cout << "6) Использует ли интернет" << endl;
cout << "7) Вернуться назад" << endl;
cout << "=====" << endl;
cout << "Ваш выбор: ";
cin >> choice3;
cout << endl;

```

```

if (choice3 < 1 || choice3 >= 7)
{
    cout << "Возвращение назад." << endl;
    break;
}

```

```

it = map.begin();
result = 0, sum = 0;
cout << "Введите критерий: ";
cin.ignore();
getline(cin, data);
number = 0, value = 0;

```

```

while (number < map.size())
{
    result = it->second->countElement(choice3, data);
    number++;
    it++;
    sum += result;
}
if (sum != 0)
    cout << "Количество элементов с данным параметром: " << sum <<

```

endl;

break;

case 4:

```

cout << "Выберите критерий, по которому надо искать: " << endl;
cout << "1) Название" << endl;
cout << "2) Время работы" << endl;
cout << "3) Размер" << endl;
cout << "4) Количество строк кода" << endl;
cout << "5) Индекс" << endl;
cout << "6) Использует ли интернет" << endl;
cout << "7) Вернуться назад" << endl;

```

```

        cout << "======" << endl;
        cout << "Ваш выбор: ";
        cin >> choise3;
        cout << endl;

        if (choise3 < 1 || choise3 >= 7)
        {
            cout << "Возвращение назад." << endl;
            break;
        }

        it = map.begin();
        cout << "Введите критерий: ";
        cin.ignore();
        getline(cin, data);
        number = 0, value = 0;

        while (number < map.size())
        {
            result = it->second->elementOutput(choise3, data);
            number++;
            it++;
        }

        break;
    case 5:
        cout << "Возвращение назад." << endl;
        break;

    default:
        cout << "Неверный символ. Повторите попытку." << endl;
        break;
}
break;

case 2:
    cout << "Введите номер элемента, который хотите удалить: ";
    cin >> value;
    cout << endl;

    findEl = 0;
    it = map.find(value);

    if (it != map.end())
    {
        map.erase(it);
        cout << "Удаление выполнено." << endl;
    }
    else
        cout << "Элемент не найден." << endl;

    break;

case 3:
    cout << "Выберите программу, которую хотите добавить:" << endl;
    cout << "1. Элемент класса CProgram" << endl;
    cout << "2. Элемент класса CMalware" << endl;
    cout << "======" << endl;
    cout << "Ваш выбор: ";
    cin >> value;

    try
    {
        if (value == 1 || value == 2)
        {
            map.emplace(++i, newProgram(value));
            cout << "Элемент добавлен." << endl;
        }
    }

```

```

        else
            cout << "Ошибка. Неверный номер." << endl;
    }
    catch (const std::exception & ex)
    {
        cout << ex.what() << endl;
    }

    break;

case 4:
    cout << "Завершение работы." << endl << endl;
    stop = 0;
    break;

default:
    cout << "Неверный символ. Повторите попытку." << endl;
    break;
}
}

void SetMenu()
{
    set <unique_ptr<CProgram>> set;
    stringstream temp;
    string data;
    bool stop = 1, findEl = 0;
    int chose = 0, chose2 = 0, chose3 = 0;
    int value = 0, number = 0, result = 0, sum = 0;
    auto it = set.begin();

    for (size_t i = 0; i < 4; i++)
    {
        if (i == 0)
            set.emplace(new CProgram());
        else if (i == 1)
            set.emplace(new CMalware(1, 8800, 555, 35, 35634, "BestMalware", "Exploit"));
        else if (i == 2)
            set.emplace(new CProgram(0, 423, 523, 654, 53453, "Calculator"));
        else if (i == 3)
            set.emplace(new CMalware(0, 345, 789, 423, 67456, "MoneyStealer", "Rootkit"));
    }

    while (stop != 0)
    {
        if (set.size() == 0)
        {
            cout << "Вектор пуст. Что вы хотите сделать?" << endl;
            cout << "1) Добавить элемент" << endl;
            cout << "2) Завершение работы" << endl;
            cout << "=====" << endl;
            cout << "Ваш выбор: ";
            cin >> chose;
            cout << endl;

            switch (chose)
            {
            case 1:
                cout << "Выберите программу, которую хотите добавить:" << endl;
                cout << "1. Элемент класса CProgram" << endl;
                cout << "2. Элемент класса CMalware" << endl;
                cout << "=====" << endl;
                cout << "Ваш выбор: ";
                cin >> value;

                try
                {
                    if (value == 1 || value == 2)

```



```

        {
            set.emplace(newProgram(value));
            cout << "Элемент добавлен." << endl;
        }
        else
            cout << "Ошибка. Неверный номер." << endl;
    }
    catch (const std::exception & ex)
    {
        cout << ex.what() << endl;
    }

    break;

case 2:
    cout << "Завершение работы." << endl;
    stop = 0;
    break;

default:
    cout << "Неверный номер элемента. Повторите попытку." << endl;
    break;
}
else
{
    cout << endl;
    cout << "1) Вывод на экран" << endl;
    cout << "2) Удаление элемента" << endl;
    cout << "3) Добавление элементов" << endl;
    cout << "4) Завершение работы" << endl;
    cout << "=====" << endl;
    cout << "Ваш выбор: ";
    cin >> choice;
    cout << endl;
}

switch (choice)
{
case 1:
    cout << "Выберите команду:" << endl;
    cout << "1) Вывести весь список на экран" << endl;
    cout << "2) Вывести программу по ID" << endl;
    cout << "3) Вывести количество элементов по критерию" << endl;
    cout << "4) Найти элемент по критерию" << endl;
    cout << "5) Вернуться к выбору действий" << endl;
    cout << "=====" << endl;
    cout << "Ваш выбор: ";
    cin >> choice2;
    cout << endl;

    switch (choice2)
    {
case 1:
        cout << setw(12) << "Название" << setw(14) << "Индекс";
        cout << setw(14) << "Время работы" << setw(8) << "Размер";
        cout << setw(18) << "Количество линий" << setw(10) << "Интернет";
        cout << setw(10) << "Тип" << endl;

        number = 1;
        for_each(set.begin(), set.end(), [&number](const unique_ptr<CProgram>&
program)
        {
            cout << number << ". " << *program << endl;
            number++;
        });
        number = 1;
        break;

```

```

case 2:
    cout << "Введите id элемента, которого вы хотите получить: ";
    cin >> value;
    cout << endl;

    findEl = 0, number = -1;
    for (const auto& element : set)
    {
        if (element->getID() == value)
        {
            number++;
            findEl = 1;
            break;
        }
        else
            number++;
    }

    if (findEl)
    {
        it = set.begin();
        advance(it, number);

        temp = (*it)->getStr();
        data = temp.str();

        cout << "Ваш элемент: " << endl;
        cout << data << endl << endl;
    }
    else
        cout << "Элемент с таким ID не найден." << endl;

    break;

case 3:
    cout << "Выберите критерий, по которому надо искать: " << endl;
    cout << "1) Название" << endl;
    cout << "2) Время работы" << endl;
    cout << "3) Размер" << endl;
    cout << "4) Количество строк кода" << endl;
    cout << "5) Индекс" << endl;
    cout << "6) Использует ли интернет" << endl;
    cout << "7) Вернуться назад" << endl;
    cout << "===== " << endl;
    cout << "Ваш выбор: ";
    cin >> choice3;
    cout << endl;

    if (choice3 < 1 || choice3 >= 7)
    {
        cout << "Возвращение назад." << endl;
        break;
    }

    it = set.begin();
    result = 0, sum = 0;
    cout << "Введите критерий: ";
    cin.ignore();
    getline(cin, data);
    number = 0, value = 0;

    while (number < set.size())
    {
        result = (*it)->countElement(choice3, data);
        number++;
        it++;
        sum += result;
    }
    if (sum != 0)

```

```

        cout << "Количество элементов с данным параметром: " << sum <<
endl;

        break;

case 4:
    cout << "Выберите критерий, по которому надо искать: " << endl;
    cout << "1) Название" << endl;
    cout << "2) Время работы" << endl;
    cout << "3) Размер" << endl;
    cout << "4) Количество строк кода" << endl;
    cout << "5) Индекс" << endl;
    cout << "6) Использует ли интернет" << endl;
    cout << "7) Вернуться назад" << endl;
    cout << "======" << endl;
    cout << "Ваш выбор: ";
    cin >> choise3;
    cout << endl;

    if (choise3 < 1 || choise3 >= 7)
    {
        cout << "Возвращение назад." << endl;
        break;
    }

    it = set.begin();
    cout << "Введите критерий: ";
    cin.ignore();
    getline(cin, data);
    number = 0, value = 0;

    while (number < set.size())
    {
        result = (*it)->elementOutput(choise3, data);
        number++;
        it++;
    }

    break;
case 5:
    cout << "Возвращение назад." << endl;
    break;

default:
    cout << "Неверный символ. Повторите попытку." << endl;
    break;

}
break;

case 2:
    cout << "Введите ID элемента, который хотите удалить: ";
    cin >> value;
    cout << endl;

    findEl = 0, number = -1;
    for (const auto& element : set)
    {
        if (element->getID() == value)
        {
            number++;
            findEl = 1;
            break;
        }
        else
            number++;
    }

    if (findEl)

```

```

        {
            it = set.begin();
            advance(it, number);
            set.erase(it);

            cout << "Удаление выполнено." << endl;
        }
        else
            cout << "Элемент не найден." << endl;

        break;

    case 3:
        cout << "Выберите программу, которую хотите добавить:" << endl;
        cout << "1. Элемент класса CProgram" << endl;
        cout << "2. Элемент класса CMalware" << endl;
        cout << "===== " << endl;
        cout << "Ваш выбор: ";
        cin >> value;

        try
        {
            if (value == 1 || value == 2)
            {
                set.emplace(newProgram(value));
                cout << "Элемент добавлен." << endl;
            }
            else
                cout << "Ошибка. Неверный номер." << endl;
        }
        catch (const std::exception & ex)
        {
            cout << ex.what() << endl;
        }

        break;

    case 4:
        cout << "Завершение работы." << endl << endl;
        stop = 0;
        break;

    default:
        cout << "Неверный символ. Повторите попытку." << endl;
        break;
    }
}
}

```

malware.cpp

```

#include "malware.h"
stringstream CMalware::getStr() const
{
    stringstream temp;

    temp << name << " " << index << " " << timeOfWork
        << " " << size << " " << amountOfLines << " "
        << useInternet << " " << type;

    return temp;
}
string CMalware::getInfo() const
{
    stringstream temp;

    temp.setf(ios::left);

```

```

temp << setw(18) << name << setw(12) << index
    << setw(11) << timeOfWork << setw(13) << size
    << setw(12) << amountOfLines << setw(12) << boolalpha << useInternet
    << setw(14) << type;

return temp.str();
}
int CMalware::countElement(int value, string data)
{
    try
    {
        if (value == 1)
        {
            if (this->name == data)
                return 1;
            else
                return 0;
        }
        else if (value == 2)
        {
            int number = stoi(data);
            if (this->timeOfWork == number)
                return 1;
            else
                return 0;
        }
        else if (value == 3)
        {
            int number = stoi(data);
            if (this->size == number)
                return 1;
            else
                return 0;
        }
        else if (value == 4)
        {
            int number = stoi(data);
            if (this->amountOfLines == number)
                return 1;
            else
                return 0;
        }
        else if (value == 5)
        {
            int number = stoi(data);
            if (this->index == number)
                return 1;
            else
                return 0;
        }
        else if (value == 6)
        {
            int number = 0;
            if (data == "true" || data == "true" || data == "1")
                number = 1;
            else
                number = 0;

            if (this->useInternet == number)
                return 1;
            else
                return 0;
        }
        else if (value == 7)
        {
            if (this->type == data)
                return 1;
            else
                return 0;
        }
    }
}

```

```

    }
}
catch (const std::exception & ex)
{
    cout << ex.what() << endl;
    return 0;
}

return 0;
}

bool CMalware::elementOutput(int value, string data)
{
    try
    {
        if (value == 1)
        {
            if (this->name == data)
                cout << *this << endl;
            return true;
        }
        else if (value == 2)
        {
            int number = stoi(data);
            if (this->timeOfWork == number)
                cout << *this << endl;
            return true;
        }
        else if (value == 3)
        {
            int number = stoi(data);
            if (this->size == number)
                cout << *this << endl;
            return true;
        }
        else if (value == 4)
        {
            int number = stoi(data);
            if (this->amountOfLines == number)
                cout << *this << endl;
            return true;
        }
        else if (value == 5)
        {
            int number = stoi(data);
            if (this->index == number)
                cout << *this << endl;
            return true;
        }
        else if (value == 6)
        {
            int number = 0;
            if (data == "true" || data == "true" || data == "1")
                number = 1;
            else
                number = 0;

            if (this->useInternet == number)
                return 1;
            else
                return 0;
        }
        else if (value == 7)
        {
            if (this->type == data)
                cout << *this << endl;
            return true;
        }
    }
}
catch (const std::exception & ex)

```

```

    {
        cout << ex.what() << endl;
        return 0;
    }

    return 0;
}

CMalware::CMalware(bool internet, int time, int size, int lines, int index, string name, string
type) : CProgram(internet, time, size, lines, index, name), type(type) {}
CMalware::CMalware() : CProgram(), type("Exploit") {}
CMalware::CMalware(const CMalware& other) : CProgram(other), type(other.type) {}
CMalware::~CMalware() {}

bool CMalware::operator==(const int id) const
{
    return this->index == id;
}

```

program.cpp

```

#include "program.h"

string CProgram::getInfo() const
{
    stringstream temp;

    temp.setf(std::ios::left);
    temp << setw(18) << name << setw(12) << index << setw(11)
        << timeOfWork << setw(13) << size << setw(12)
        << amountOfLines << setw(8) << boolalpha << useInternet;

    return temp.str();
}
int CProgram::getID() const
{
    return index;
}
stringstream CProgram::getStr() const
{
    stringstream temp;
    temp << name << " " << index << " " << timeOfWork << " "
        << size << " " << amountOfLines << " " << useInternet;

    return temp;
}
int CProgram::countElement(int value, string data)
{
    try
    {
        if (value == 1)
        {
            if (this->name == data)
                return 1;
            else
                return 0;
        }
        else if (value == 2)
        {
            int number = stoi(data);
            if (this->timeOfWork == number)
                return 1;
            else
                return 0;
        }
        else if (value == 3)

```

```

{
    int number = stoi(data);
    if (this->size == number)
        return 1;
    else
        return 0;
}
else if (value == 4)
{
    int number = stoi(data);
    if (this->amountOfLines == number)
        return 1;
    else
        return 0;
}
else if (value == 5)
{
    int number = stoi(data);
    if (this->index == number)
        return 1;
    else
        return 0;
}
else if (value == 6)
{
    int number = 0;
    if (data == "true" || data == "true" || data == "1")
        number = 1;
    else
        number = 0;

    if (this->useInternet == number)
        return 1;
    else
        return 0;
}
}
catch (const std::exception& ex)
{
    cout << ex.what() << endl;
    return 0;
}

return 0;
}

bool CProgram::elementOutput(int value, string data)
{
    try
    {
        if (value == 1)
        {
            if (this->name == data)
                cout << *this << endl;
            return true;
        }
        else if (value == 2)
        {
            int number = stoi(data);
            if (this->timeOfWork == number)
                cout << *this << endl;
            return true;
        }
        else if (value == 3)
        {
            int number = stoi(data);
            if (this->size == number)
                cout << *this << endl;
            return true;
        }
    }
}

```



```

        else if (value == 4)
        {
            int number = stoi(data);
            if (this->amountOfLines == number)
                cout << *this << endl;
            return true;
        }
        else if (value == 5)
        {
            int number = stoi(data);
            if (this->index == number)
                cout << *this << endl;
            return true;
        }
        else if (value == 6)
        {
            int number = 0;
            if (data == "true" || data == "true" || data == "1")
                number = 1;
            else
                number = 0;

            if (this->useInternet == number)
                return 1;
            else
                return 0;
        }
    }
    catch (const std::exception& ex)
    {
        cout << ex.what() << endl;
        return 0;
    }

    return 0;
}

ostream& operator<< (ostream& output, const CProgram& program)
{
    output << program.getInfo();
    return output;
}

bool CProgram::operator==(const int id) const
{
    return this->index == id;
}

CProgram::CProgram(bool internet, int time, int size, int lines, int index, string name) :
    useInternet(internet), timeOfWork(time), size(size), amountOfLines(lines), index(index), name(name)
{}

CProgram::CProgram() : useInternet(false), timeOfWork(0), size(0), amountOfLines(0), index(0101),
    name("Basic") {}

CProgram::CProgram(const CProgram& other) : useInternet(other.useInternet),
    timeOfWork(other.timeOfWork), size(other.size), amountOfLines(other.amountOfLines),
    index(other.index), name(other.name) {}

CProgram::~CProgram() {}

```

test.cpp

```
#include "malware.h"

void VectorTest();
void ListTest();
void MapTest();
void SetTest();

int main()
{
    setlocale(LC_ALL, "Rus");

    VectorTest();
    ListTest();
    MapTest();
    SetTest();

    if (_CrtDumpMemoryLeaks())
        cout << "\nЕсть утечка памяти.\n";
    else
        cout << "\nУтечка памяти отсутствует.\n";

    return 0;
}

void VectorTest()
{
    cout << "Vector" << endl;
    vector<unique_ptr<CProgram>> vector;
    std::vector<unique_ptr<CProgram>>::const_iterator it;
    stringstream line;
    string data;
    int vectorSize;
    int value, result = 0, sum = 0;
    int i = 0;

    for (size_t i = 0; i < 4; i++)
    {
        if (i == 0)
            vector.emplace_back(new CProgram());
        else if (i == 1)
            vector.emplace_back(new CMalware(1, 8800, 555, 35, 35634, "BestMalware", "Exploit"));
        else if (i == 2)
            vector.emplace_back(new CProgram(0, 423, 523, 654, 53453, "Calculator"));
        else if (i == 3)
            vector.emplace_back(new CMalware(0, 345, 789, 423, 67456, "MoneyStealer", "Rootkit"));
    }

    vectorSize = vector.size();
    vector.emplace_back(new CMalware());
    if(vectorSize != vector.size())
        cout << "Тест добавления элемента\tвыполнен успешно.\n";
    else
        cout << "Тест добавления элемента\tне выполнен успешно.\n";

    it = vector.begin();
    advance(it, 2);
    vector.erase(it);
    if (vectorSize == vector.size())
        cout << "Тест удаления элемента\t\tвыполнен успешно.\n";
    else
        cout << "Тест удаления элемента\t\tне выполнен успешно.\n";

    line = vector[0]->getStr();
    data = line.str();
    if (data == "Basic 65 0 0 0 0")
        cout << "Тест получения элемента\t\tвыполнен успешно.\n";
    else
        cout << "Тест получения элемента\t\tне выполнен успешно.\n";
}
```

```

it = vector.begin();
data = "false";
while (i < vector.size())
{
    result = (*it)->countElement(6, data);
    i++;
    it++;
    sum += result;
}
if (sum == 3)
    cout << "Тест подсчёта элементов\t\tвыполнен успешно.\n";
else
    cout << "Тест подсчёта элементов\t\tне выполнен успешно.\n";
}
void ListTest()
{
    cout << endl << "List" << endl;
    list<unique_ptr<CProgram>> list;
    std::list<unique_ptr<CProgram>>::const_iterator it;
    int listSize;
    int value, sum = 0, result = 0;
    int i = 0;
    stringstream line;
    string data;

    for (size_t i = 0; i < 4; i++)
    {
        if (i == 0)
            list.emplace_back(new CProgram());
        else if (i == 1)
            list.emplace_back(new CMalware(1, 8800, 555, 35, 35634, "BestMalware", "Exploit"));
        else if (i == 2)
            list.emplace_back(new CProgram(0, 423, 523, 654, 53453, "Calculator"));
        else if (i == 3)
            list.emplace_back(new CMalware(0, 345, 789, 423, 67456, "MoneyStealer", "Rootkit"));
    }

    listSize = list.size();
    list.emplace_back(new CMalware());
    if (listSize < list.size())
        cout << "Тест добавления элемента\t\tвыполнен успешно.\n";
    else
        cout << "Тест добавления элемента\t\tне выполнен успешно.\n";

    it = list.begin();
    list.erase(it);
    if (list.size() == listSize)
        cout << "Тест удаления элемента\t\tвыполнен успешно.\n";
    else
        cout << "Тест удаления элемента\t\tне выполнен успешно.\n";

    it = list.begin();
    line = (*it)->getStr();
    data = line.str();
    if (data == "BestMalware 35634 8800 555 35 1 Exploit")
        cout << "Тест получения элемента\t\tвыполнен успешно.\n";
    else
        cout << "Тест получения элемента\t\tне выполнен успешно.\n";

    it = list.begin();
    data = "false";
    while (i < list.size())
    {
        result = (*it)->countElement(6, data);
        i++;
        it++;
        sum += result;
    }
}

```

```

if (sum == 3)
    cout << "Тест подсчёта элементов\t\tвыполнен успешно.\n";
else
    cout << "Тест подсчёта элементов\t\tне выполнен успешно.\n";
}
void SetTest()
{
    cout << endl << "Set" << endl;
    set <unique_ptr<CProgram>> set;
    std::set<unique_ptr<CProgram>>::const_iterator it;
    stringstream line;
    string data;
    int setSize, sum = 0, i = 0;
    int value;

    for (size_t i = 0; i < 4; i++)
    {
        if (i == 0)
            set.emplace(new CProgram());
        else if (i == 1)
            set.emplace(new CMalware(1, 8800, 555, 35, 35634, "BestMalware", "Exploit"));
        else if (i == 2)
            set.emplace(new CProgram(0, 423, 523, 654, 53453, "Calculator"));
        else if (i == 3)
            set.emplace(new CMalware(0, 345, 789, 423, 67456, "MoneyStealer", "Rootkit"));
    }

    setSize = set.size();
    set.emplace(new CMalware());
    if (setSize < set.size())
        cout << "Тест добавления элемента\t\tвыполнен успешно.\n";
    else
        cout << "Тест добавления элемента\t\tне выполнен успешно.\n";

    it = set.begin();
    set.erase(it);
    if (set.size() == setSize)
        cout << "Тест удаления элемента\t\tвыполнен успешно.\n";
    else
        cout << "Тест удаления элемента\t\tне выполнен успешно.\n";

    it = set.begin();
    line = (*it)->getStr();
    data = line.str();
    if (data == "MoneyStealer 67456 345 789 423 0 Rootkit" || data == "BestMalware 35634 8800 555 35
1 Exploit" || data == "Basic 65 0 0 0 0 Exploit")
        cout << "Тест получения элемента\t\tвыполнен успешно.\n";
    else
        cout << "Тест получения элемента\t\tне выполнен успешно.\n";

    it = set.begin();
    data = "53453";
    while (i < set.size())
    {
        value = (*it)->countElement(5, data);
        i++;
        it++;
        sum += value;
    }
    if (sum == 1)
        cout << "Тест подсчёта элементов\t\tвыполнен успешно.\n";
    else
        cout << "Тест подсчёта элементов\t\tне выполнен успешно.\n";
}
void MapTest()
{
    cout << endl << "Map" << endl;
    std::map<int, unique_ptr<CProgram>>::const_iterator it;
    map <int, unique_ptr<CProgram>> map;

```

```

stringstream line;
string data;
int number = 0, result, sum = 0;
int mapSize;
int i = 0;

for (; i < 4; i++)
{
    if (i == 0)
        map.emplace(i + 1, new CProgram());
    else if (i == 1)
        map.emplace(i + 1, new CMalware(1, 8800, 555, 35, 35634, "BestMalware", "Exploit"));
    else if (i == 2)
        map.emplace(i + 1, new CProgram(0, 423, 523, 654, 53453, "Calculator"));
    else if (i == 3)
        map.emplace(i + 1, new CMalware(0, 345, 789, 423, 67456, "MoneyStealer", "Rootkit"));
}

mapSize = map.size();
map.emplace(++i, new CMalware);
if (mapSize < map.size())
    cout << "Тест добавления элемента\tвыполнен успешно.\n";
else
    cout << "Тест добавления элемента\tне выполнен успешно.\n";

it = map.begin();
map.erase(it);
if (mapSize == map.size())
    cout << "Тест удаления элемента\t\tвыполнен успешно.\n";
else
    cout << "Тест удаления элемента\t\tне выполнен успешно.\n";

it = map.begin();
line = it->second->getStr();
data = line.str();
if (data == "BestMalware 35634 8800 555 35 1 Exploit")
    cout << "Тест получения элемента\t\tвыполнен успешно.\n";
else
    cout << "Тест получения элемента\t\tне выполнен успешно.\n";

data = "53453";
while (number < map.size())
{
    result = it->second->countElement(5, data);
    number++;
    it++;
    sum += result;
}
if (sum == 1)
    cout << "Тест подсчёта элементов\t\tвыполнен успешно.\n";
else
    cout << "Тест подсчёта элементов\t\tне выполнен успешно.\n";
}

```

Header.h

```
#pragma once
#define _CRT_SECURE_NO_WARNINGS
#define CRTDBG_MAP_ALLOC
#include <crtdbg.h>
#define DEBUG_NEW new(_NORMAL_BLOCK, FILE, __LINE)

#include <string>
#include <iostream>
#include <iomanip>
#include <locale>
#include <fstream>
#include <sstream>
#include <istream>
#include <vector>
#include <memory>
#include <list>
#include <map>
#include <set>
#include <algorithm>

using std::string;
using std::cin;
using std::cout;
using std::endl;
using std::setw;
using std::boolalpha;
using std::setiosflags;
using std::ios;
using std::ifstream;
using std::ostream;
using std::ofstream;
using std::stringstream;
using std::istream;
using std::vector;
using std::list;
using std::map;
using std::set;
using std::unique_ptr;
using std::advance;
using std::stoi;
using std::for_each;
```

malware.h

```
#pragma once
#include "program.h"
class CMalware final: public CProgram
{
private:
    string type;

public:
    string getInfo() const override final;
    stringstream getStr() const override final;
    bool elementOutput(int, string) override final;
    int countElement(int, string) override final;

    CMalware();
    CMalware(bool, int, int, int, int, string, string);
    CMalware(const CMalware&);
    ~CMalware() override final;

    bool operator==(const int) const override final;
};
```

program.h

```
#pragma once
#include "Header.h"

class CProgram
{ protected:
    int timeOfWork;        //average time of program execution
    int size;              //size of program
    int amountOfLines;     //number of lines in code
    int index;             //index
    bool useInternet;      //use internet
    string name;           //name of program

public:
    virtual string getInfo() const;
    virtual stringstream getStr() const;
    int getID() const;
    virtual bool elementOutput(int, string);
    virtual int countElement(int, string);

    CProgram();
    CProgram(bool, int, int, int, int, string);
    CProgram(const CProgram&);
    virtual ~CProgram();

    friend ostream& operator<< (ostream&, const CProgram&);
    virtual bool operator==(const int) const;
};
```

4. Результати роботи про грами

```
Виберите команду:
1) Вывести весь список на экран
2) Вывести программу по ID
3) Вывести количество элементов по критерию
4) Найти элемент по критерию
5) Вернуться к выбору действий
=====
Ваш выбор: 1

   Название      Индекс  Время работы  Размер  Количество линий  Интернет  Тип
1. Basic         65      0             555     0               false     Exploit
2. BestMalware   35634   8800          555     35              true      Exploit
3. Calculator     53453   423           523     654             false     Rootkit
4. MoneyStealer   67456   345           789     423             false     Rootkit

1) Вывод на экран
2) Удаление элемента
3) Добавление элементов
4) Завершение работы
=====
Ваш выбор: 1

Виберите команду:
1) Вывести весь список на экран
2) Вывести программу по ID
3) Вывести количество элементов по критерию
4) Найти элемент по критерию
5) Вернуться к выбору действий
=====
Ваш выбор: 3

Виберите критерий, по которому надо искать:
1) Название
2) Время работы
3) Размер
4) Количество строк кода
5) Индекс
6) Использует ли интернет
7) Вернуться назад
=====
Ваш выбор: 6

Введите критерий: false
Количество элементов с данным параметром: 3

1) Вывод на экран
2) Удаление элемента
3) Добавление элементов
4) Завершение работы
=====
Ваш выбор: 1

Виберите команду:
1) Вывести весь список на экран
2) Вывести программу по ID
3) Вывести количество элементов по критерию
4) Найти элемент по критерию
5) Вернуться к выбору действий
=====
Ваш выбор: 5

Возвращение назад.

1) Вывод на экран
2) Удаление элемента
3) Добавление элементов
4) Завершение работы
=====
Ваш выбор: 4

Завершение работы.

Виберите STL контейнер:
1. Vector
2. List
3. Map
4. Set
5. Выход
=====
Ваш выбор: 5

Утечка памяти отсутствует.
```

```
Vector
Тест добавления элемента      выполнен успешно.
Тест удаления элемента        выполнен успешно.
Тест получения элемента       выполнен успешно.
Тест подсчёта элементов       выполнен успешно.

List
Тест добавления элемента      выполнен успешно.
Тест удаления элемента        выполнен успешно.
Тест получения элемента       выполнен успешно.
Тест подсчёта элементов       выполнен успешно.

Map
Тест добавления элемента      выполнен успешно.
Тест удаления элемента        выполнен успешно.
Тест получения элемента       выполнен успешно.
Тест подсчёта элементов       выполнен успешно.

Set
Тест добавления элемента      выполнен успешно.
Тест удаления элемента        выполнен успешно.
Тест получения элемента       выполнен успешно.
Тест подсчёта элементов       выполнен успешно.

Утечка памяти отсутствует.
```

5. Висновки

При виконанні даної лабораторної роботи було набуто практичного досвіду роботи з STL контейнерами переміщення та пошуку.

Програма протестована, витоків пам'яті немає, виконується без помилок.