

Лабораторна робота 12. STL

Тема: STL. Ітератори. Послідовні контейнери. Цикл *range-for*. Асоціативні контейнери.

Мета: отримати базові знання про STL-контейнери. Освоїти основні механізми роботи з STL контейнерами.

1. Завдання до роботи

Загальне завдання. Маючи класи з прикладної області РГЗ (тільки базовий клас та клас/класи спадкоємці), створити діалогове меню, що дозволяє продемонструвати роботу STL-контейнерів (додавання / видалення / отримання даних, показ усіх елементів) та показати їх принципову різницю:

- vector;
- set;
- list;
- map .

При цьому врахувати, що контейнери містять елементи одного типу, наприклад, базового.

Прохід по всьому контейнеру повинен виконуватися за допомогою циклу мови C++11 – range-for.

Додаткове завдання на оцінку «відмінно»:

- контейнери повинні оперувати даними не тільки базового класу, а ще даними класів-спадкоємців.

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`).

`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 "Header.h"
#include "program.h"
#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;
    stringstream temp;
    string data;
    bool stop = 1, findEl = 0;
    int chose = 0, chose2 = 0;
    int value = 0;
    int number = 0;
    auto it = vector.begin();

    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) 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 (const auto& element:vector)
    {
        cout << number << ". " << *element << 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:
    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 chose = 0, chose2 = 0;
    int value = 0;
    int number = 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 >> choice;
            cout << endl;

            switch (choice)
            {
            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 >> chose;
        cout << endl;
    }

    switch (chose)
    {
    case 1:
        cout << "Выберите команду:" << endl;
        cout << "1) Вывести весь список на экран" << endl;
        cout << "2) Вывести программу по ID" << endl;
        cout << "3) Вернуться к выбору действий" << endl;
        cout << "===== " << endl;
        cout << "Ваш выбор: ";
        cin >> chose2;
        cout << endl;

        switch (chose2)
        {
        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 (const auto& element : list)
            {
                cout << number << ". " << *element << 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:
            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;
    int value = 0;
    int i = 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 >> 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) 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;

                for (const auto& element : map)
                    cout << element.first << ". " << *element.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:
        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;
int value = 0;
int number = 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 >> 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) 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 (const auto& element : set)
        {
            cout << number << ". " << *element << 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:
    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();
}

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;
}

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)
{
    //cout << "\nВызвался конструктор с параметрами";
}

CProgram::CProgram() : useInternet(false), timeOfWork(0), size(0), amountOfLines(0), index(0101),
    name("Basic")
{
    //cout << "\nВызвался конструктор по умолчанию.";
}

CProgram::CProgram(const CProgram& other) : useInternet(other.useInternet),
    timeOfWork(other.timeOfWork), size(other.size), amountOfLines(other.amountOfLines),
    index(other.index), name(other.name)
{
    //cout << "\nВызвался конструктор копирования.";
}

CProgram::~CProgram()
{
    //cout << "\nВызвался деструктор";
}
```

test.cpp

```
#include "Header.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()
{
    vector<int> vector = { 1, -5, 20, 555, 0 };
    int vectorSize = vector.size();
    int newVectorSize;
    int value;
    std::vector<int>::iterator it;
    cout << "Vector" << endl;

    vector.push_back(155);
    newVectorSize = vector.size();
    if(vectorSize != newVectorSize && vector[newVectorSize - 1] == 155)
        cout << "Тест добавления элемента\tвыполнен успешно.\n";
    else
        cout << "Тест добавления элемента\tне выполнен успешно.\n";

    it = vector.begin();
    value = vector[2];
    vector.erase(it + 2);
    newVectorSize = vector.size();
    if (vectorSize == newVectorSize && vector[2] != value)
        cout << "Тест удаления элемента\t\tвыполнен успешно.\n";
    else
        cout << "Тест удаления элемента\t\tне выполнен успешно.\n";

    if (vector[0] == 1)
        cout << "Тест получения элемента\t\tвыполнен успешно.\n";
    else
        cout << "Тест получения элемента\t\tне выполнен успешно.\n";
}

void ListTest()
{
    list<int> list = { 1, -5, 20, 555, 0 };
    int listSize = list.size();
    int value;
    std::list<int>::iterator it;
    std::list<int>::iterator it2;
    cout << endl << "List" << endl;

    list.push_back(155);
    list.push_front(228);
    it = list.begin();
    it2 = list.begin();
    std::advance(it2, list.size() - 1);
```

```

if (listSize != list.size() && *it == 228 && *it2 == 155)
    cout << "Тест добавления элемента\tвыполнен успешно.\n";
else
    cout << "Тест добавления элемента\tне выполнен успешно.\n";

it2 = list.begin();
std::advance(it2, 2);
list.erase(it2);
it = list.begin();
std::advance(it, 2);
if (list.size() == listSize+1 && it != it2)
    cout << "Тест удаления элемента\t\tвыполнен успешно.\n";
else
    cout << "Тест удаления элемента\t\tне выполнен успешно.\n";

if (*it == 20)
    cout << "Тест получения элемента\t\tвыполнен успешно.\n";
else
    cout << "Тест получения элемента\t\tне выполнен успешно.\n";
}
void SetTest()
{
    set<int> set = { 1, -5, 20, 555, 0 };
    int setSize = set.size();
    int value;
    std::set<int>::iterator it;
    std::set<int>::iterator it2;
    cout << endl << "Set" << endl;

    set.insert(155);
    it2 = set.begin();
    std::advance(it2, 4);
    if (setSize != set.size() && *it2 == 155)
        cout << "Тест добавления элемента\tвыполнен успешно.\n";
    else
        cout << "Тест добавления элемента\tне выполнен успешно.\n";

    it2 = set.begin();
    set.erase(it2);
    it = set.begin();
    if (set.size() == setSize && it != it2 && *it == 0)
        cout << "Тест удаления элемента\t\tвыполнен успешно.\n";
    else
        cout << "Тест удаления элемента\t\tне выполнен успешно.\n";

    if (*it == 0)
        cout << "Тест получения элемента\t\tвыполнен успешно.\n";
    else
        cout << "Тест получения элемента\t\tне выполнен успешно.\n";
}
void MapTest()
{
    map<int, int> map = { {1, 1}, {-5, 2}, {20, 3}, {555, 4}, {0, 5} };
    int mapSize = map.size();
    std::map<int, int>::iterator it;
    std::map<int, int>::iterator it2;
    cout << endl << "Map" << endl;

    map.insert(std::pair<int, int>(155, 6));
    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();
    if (map.find(0) == it)
        cout << "Тест получения элемента\t\tвыполнен успешно.\n";
    else
        cout << "Тест получения элемента\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>

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;

```

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;

    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;

    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. Результаты работы про грами

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

Завершение работы.

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

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

Выберите команду:
1) Вывести весь список на экран
2) Вывести программу по ID
3) Вернуться к выбору действий
=====
Ваш выбор: 1

      Название      Индекс  Время работы  Размер  Количество линий  Интернет  Тип
1. Basic            65        0           0         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: 2

Введите номер элемента, который хотите удалить: 3

Удаление выполнено.

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

Выберите команду:
1) Вывести весь список на экран
2) Вывести программу по ID
3) Вернуться к выбору действий
=====
Ваш выбор: 1

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

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

Завершение работы.

Выберите 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.

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

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

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

Утечка памяти отсутствует.
```

5. Висновки

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

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