

ЛАБОРАТОРНАЯ РАБОТА №2

«ИССЛЕДОВАНИЕ МЕХАНИЗМА ВИРТУАЛЬНЫХ ФУНКЦИЙ»

Цель работы: Приобретение практических навыков при написании объектно-ориентированных программ с использованием механизмов наследования и виртуальных функций. Освоение особенностей отладки объектно-ориентированных программ.

Вариант задания

Для заданной по варианту иерархии описать классы, конструкторы и деструктор, функции ввода и вывода информации на экран. Базовый класс определить как абстрактный, а заданную функцию — как чисто виртуальную в базовом классе и переопределить ее в остальных классах иерархии. Проиллюстрировать корректную работу виртуальных функций и механизма наследования.

Вариант 11

Создать абстрактный базовый класс Поиск с виртуальной функцией поиска. Создать производные классы: символ (char), строка (char *) и матрица (char **). Функция проверяет, существует ли введенная буква в строке/матрице/равна ли символу?

2. Код программы на языке C++

```
#include <string>
#include <iostream>

using namespace std;
class Base_search
{
public:
    virtual bool find_elem(const char) = 0;
};

class char_search : public Base_search {
    const char c;
public:
    char_search(const char c) : c(c) {}
    virtual bool find_elem(const char c) override {
        return this->c == c;
    }
};

class string_search : public Base_search {
    const string s;
public:
    string_search(const string s) : s(s) {}
    virtual bool find_elem(const char c) override {
        for (char e : s)
        {
            if (e == c) return true;
        }
        return false;
    }
};

class string_list_search : public Base_search {
    string* s;
    int s_count;
public:
    string_list_search(const string* s, int s_count) {
        this->s = new string[s_count];
        for (int i = 0; i < s_count; i++)
            this->s[i] = s[i];
        this->s_count = s_count;
    }
    virtual bool find_elem(const char c) override {
        for (int i = 0; i < s_count; i++) {
            for (char e : s[i])
            {
                if (e == c) return true;
            }
        }
        return false;
    }
};

int main()
{
    system("chcp 1251");
    char_search obj1('L');
    string_search obj2("This is testovay strocka");
    cout << "Введите количествое строк в матрице ";
    int s_count;
    cin >> s_count;
    string* s = new string[s_count];
    for (int i = 0; i < s_count; i++) {
        cout << "Введите строку " << i << " -->";
```

```

        cin >> s[i];
    }
    string_list_search obj3(s, s_count); char serch_el;
    while (1) {
        cout << "Введите символ для поиска. 0 для выхода ";
        cin >> serch_el;
        if (serch_el == '0') break;
        if (obj1.find_elem(serch_el)) cout << "Данный символ есть в 1-ом
классе"<<endl;
        if (obj2.find_elem(serch_el)) cout << "Данный символ есть в 2-ом классе" <<
endl;
        if (obj3.find_elem(serch_el)) cout << "Данный символ есть в 3-ом классе" <<
endl;
    }
}

```

3. Тестирование и отладка

Для первого класса будет передан символ L, для второго класса будет передана строка **This is testovay strocka**. Для третьего класса будет сделан ввод данных с клавиатуры.

```

E:\University\ООП\Вторая лаба\Исходники\Laba2\Debug\Laba2.exe
Текущая кодовая страница: 1251
Введите количество строк в матрице 4
Введите строку 0 -->Строка
Введите строку 1 -->Тестовая
Введите строку 2 -->Это
Введите строку 3 -->Интересно
Введите символ для поиска. 0 для выхода в
Введите символ для поиска. 0 для выхода ы
Введите символ для поиска. 0 для выхода с
Введите символ для поиска. 0 для выхода т
Данный символ есть в 3-ом классе
Введите символ для поиска. 0 для выхода T
Данный символ есть в 2-ом классе
Введите символ для поиска. 0 для выхода te
Данный символ есть в 2-ом классе
Введите символ для поиска. 0 для выхода Данный символ есть в 2-ом классе
Введите символ для поиска. 0 для выхода s
Данный символ есть в 2-ом классе
Введите символ для поиска. 0 для выхода d
Введите символ для поиска. 0 для выхода s
Данный символ есть в 2-ом классе
Введите символ для поиска. 0 для выхода a
Данный символ есть в 2-ом классе
Введите символ для поиска. 0 для выхода d
Введите символ для поиска. 0 для выхода f
Введите символ для поиска. 0 для выхода g
Введите символ для поиска. 0 для выхода v
Данный символ есть в 2-ом классе
Введите символ для поиска. 0 для выхода L
Данный символ есть в 1-ом классе
Введите символ для поиска. 0 для выхода l
Введите символ для поиска. 0 для выхода das
Введите символ для поиска. 0 для выхода Данный символ есть в 2-ом классе
Введите символ для поиска. 0 для выхода Данный символ есть в 2-ом классе
Введите символ для поиска. 0 для выхода sadafghjm
Данный символ есть в 2-ом классе
Введите символ для поиска. 0 для выхода Данный символ есть в 2-ом классе
Введите символ для поиска. 0 для выхода Введите символ для поиска. 0 для выхода Данный символ есть в 2-ом классе
Введите символ для поиска. 0 для выхода Введите символ для поиска. 0 для выхода Данный символ есть в 2-ом классе
Введите символ для поиска. 0 для выхода Введите символ для поиска. 0 для выхода Введите символ для поиска. 0 для выхода

```

Рисунок 1 – Ввода стартовых данных и тестирования

После ввода данных, программа запрашивает ввод символа, после его ввода, она проверяет его, и по итогу, потом просто вызывает методы у класса.

В результате тестирования, видно, что классы работают правильно, виртуальный метод и поля наследуются корректно.

Вывод

При выполнении данной лабораторной работы были приобретены практических навыков при написании объектно-ориентированных программ с использованием механизмов наследования и виртуальных функций. Освоены особенностей отладки объектно-ориентированных программ.