|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

##### ФАКУЛЬТЕТ ИУ «Информатика и системы управления»

**КАФЕДРА ИУ8 «Компьютерная безопасность*»***

**ЛАБОРАТОРНАЯ РАБОТА №3**

**Название лабораторной работы: Наследование классов.**

**по курсу: Алгоритмические языки.**

**Группа ИУ8 – 24**

**Студент: Полонский Сергей**

**Преподаватель: Барыкин Дмитрий**

**Дата: 13.03.2024**

1. **Цель работы:** На практике научится применять наследование классов: создавать наследников класса, создавать для них конструкторы базирующиеся на классе родителе и переопределять функции классов родителей использующиеся в классе наследнике.
2. **Условие задачи:** Создать базовый класс «вещественное число». Элементы класса: поле, задающее значение числа (статус доступа *protected*); конструктор для инициализации поля; функция для вычисления модуля числа; функция для печати поля и модуля числа. Создать производный класс «комплексное число». Элементы класса: дополнительно поле, задающее значение мнимой части числа; конструктор для инициализации полей; переопределенная функция для вычисления модуля числа (модуль числа – корень квадратный из суммы квадратов вещественной и мнимой частей числа); переопределенная функция для печати полей и модуля числа. Создать по 1 объекту каждого из классов. Показать вызов созданных функций. При переопределении функций обеспечить и продемонстрировать два варианта: статический полиморфизм и динамический полиморфизм.
3. **Код:**

#include <iostream>

#include <math.h>

using namespace std;

class vesh\_chisl{ // класс вещественное число

public:

vesh\_chisl(); // конструктор, который создает число 0

vesh\_chisl(double c); // конструктор, который создает число c

double modul(); // модуль числа, будет показывать статический полиморфизм

virtual void print(); // функция вывода на экран, будет показывать динамический полиморфизм

protected:

double chislo; // число

};

vesh\_chisl::vesh\_chisl() {

chislo = 0;

}

vesh\_chisl::vesh\_chisl(double c) {

chislo = c;

}

double vesh\_chisl::modul() {

return abs(chislo);

}

void vesh\_chisl::print() {

cout << chislo << endl;

}

class complex\_chisl : public vesh\_chisl { // класс комплексное число - наследник класс вещественное число

public:

complex\_chisl(); // конструктор, который делает 0 вещественное число

complex\_chisl(double a, double b); // конструктор, который делает комплексное число с действительной частью a, и вещественной частью b

double modul(); // модуль комплексного числа

void print(); // вывод комплексного число в две строки: в первой строке действительная часть, во второй мнимая

protected:

double mnim; // мнимое число

};

complex\_chisl::complex\_chisl() {

vesh\_chisl();

mnim = 0;

};

complex\_chisl::complex\_chisl(double a, double b):

vesh\_chisl(a)

{

mnim = b;

}

double complex\_chisl::modul() { // переопределение функции modul

return pow(pow(mnim, 2) + pow(chislo, 2), 0.5);

};

void complex\_chisl::print() { // переопределение функции print

vesh\_chisl::print();

cout << mnim << endl;

};

int main()

{

vesh\_chisl\* a = new complex\_chisl(2, 3);

a->print();

cout << a->modul(); // выдаст 2, а по факту это значение равно 3.6...

cout << endl << endl << endl;

complex\_chisl\* b = new complex\_chisl(5, 6);

b->print();

cout << b->modul() << endl;

}

1. **Вывод результатов работы программы:**

2

3

2

5

6

7.81025

1. **Вывод по работе:** в работы мы научились создавать наследников класса родителя, взаимодействовать с ними, переопределять функции классов наследников, создавать для них конструкторы. Так же разобрались с понятием статический и динамический полиморфизм.