|  |  |
| --- | --- |
| Gerb-BMSTU_01 | **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 **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.03.01 Информатика и вычислительная техника**

**Отчет**

|  |  |
| --- | --- |
| **По Рубежному контролю №** | 3 |

**Название:**

Наследование С++.

**Дисциплина:** Объектно-ориентированное программирование

Вариант 5.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Студент | ИУ6-21Б |  |  | Т.Е.Старжевский |
|  | (Группа) |  | (Подпись, дата) | (И.О. Фамилия) |
|  |  |  |  |  |
| Преподаватель |  |  |  | О.А.Веселовская |
|  |  |  | (Подпись, дата) | (И.О. Фамилия) |

Москва, 2022

**Текст задания:**

Реализовать иерархию классов в соответствии с рисунком.

**Обязательные требования:**

1. Класс А – должен включать:

- конструктор, инициализирующий массив D вещественных чисел из 10-ти элементов;

- абстрактный метод Out для реализации механизма полиморфизма;

- метод Print(), содержащий обращение к методу Out для вывода массива;

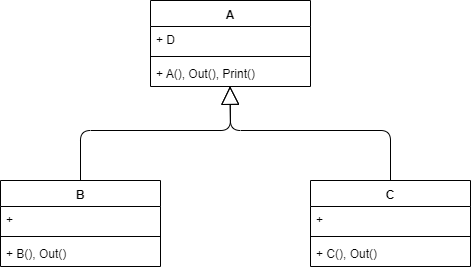
2. Класс B – должен включать метод Out, позволяющий выводить массив D на экран в строку.

3. Класс C -должен включать метод Out, который выводит элементы массива D на экран по одному в строке.

4. Продемонстрировать работу метода Print() для объектов классов B и C.

5. Изобразить подробную диаграмму классов.

**Диаграмма классов:**



**Код программы:**

#pragma once

#include <iostream>

class A

{

public:

float D[10];

A(float iD[10])

{

for (int i = 0; i < 10; i++)

{

D[i] = iD[i];

}

}

A()

{

for (int i = 0; i < 10; i++)

{

D[i] = 0;

}

}

virtual void Out() = 0;

void Print()

{

Out();

}

};

class B : public A

{

public:

B(float iD[10]) : A(iD) {};

B() : A() {};

void Out() override

{

for (int i = 0; i < 10; i++)

{

std::cout << D[i] << " ";

}

}

};

class C : public A

{

public:

C(float iD[10]) : A(iD) {};

C() : A() {};

void Out() override

{

for (int i = 0; i < 10; i++)

{

std::cout << D[i] << "\n";

}

}

};

#include <iostream>

#include "classes.h"

#include <time.h>

#include <conio.h>

using namespace std;

int main()

{

srand(time(NULL));

float mas\_float[10];

for (int i = 0; i < 10; i++)

{

mas\_float[i] = (rand() % 1000) / 10;

}

B B(mas\_float);

C C(mas\_float);

cout << "==== Class B metod Print() ====\n";

\_getch();

B.Print();

cout << "\n==== Class C metod Print() ====\n";

\_getch();

C.Print();

return 0;

}

**Результаты тестирования:**

|  |  |  |
| --- | --- | --- |
| Исходные данные | Ожидаемый результат | Полученный результат |
| 44 58 37 30 5 23 68 89 51 2 | Вывод в строку, в столбец |  |

При заданных значениях программа работает корректно.

**Вывод:** Научился программировать с использованием механизмов наследования на языке C++.