ГУАП

КАФЕДРА № 44

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Ассистент |  |  |  | Т.Р. Мустафин |
| должность, уч. степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ |
| МНОЖЕСТВЕННОЕ НАСЛЕДОВАНИЕ. ПОЛИМОРФИЗМ |
| по курсу: ОСНОВЫ ПРОГРАММИРОВАНИЯ |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4319 |  |  |  | Д.Д. Евсеев |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2024

**Цель работы:**

Изучение видов наследования и способов их применения; изучение способов использования виртуальных функций, абстрактных классов; получение навыков объектно-ориентированного программирования на языке C++.

**Индивидуальное задание**

Вариант 6

Космический аппарат – Движитель – Комбинации

**Описание выбранной модели решения**

Созданы главный файл, продемонстрировано переопределение чисто виртуального метода, созданы заголовочные файлы для каждого из классов в системе наследования: Базовый класс Spacecraft, его наследники первого уровня Space, Mover и Ignition, наследники второго уровня Polet и Move. Продемонстрировано их взаимодействие на примере простейших функций.

**Текст программы**

**//main.cpp**

#include"Vzlet.h"

#include"Mover.h"

#include"Ignition.h"

#include "Spacecraft.h"

#include"Polet.h"

#include "Move.h"

int main() {

Polet obj;

obj.height();

obj.status();

Mover obj1;

obj1.status();

obj1.height();

cout << endl;

Vzlet obj2;

obj2.height();

obj2.status();

Move rocket;

rocket.height();

rocket.status();

}

**//Spacecraft.h**

#pragma once

#include <string>

#include <iostream>

using namespace std;

class Spacecraft {

protected:

string power;

string oil;

string ready;

public:

virtual void status();

virtual void height()=0;

};

void Spacecraft::status() {

cout << "Unready";

}

//**Vzlet.h**

#pragma once

#include "Spacecraft.h"

class Vzlet : public Spacecraft {

public:

Vzlet();

void status() override;

void height() override;

~Vzlet();

};

Vzlet::Vzlet() {

power = "Very high";

oil = "Full";

ready = "Ready";

}

void Vzlet::status() {

cout << "power: " << power << endl;

cout << "oil: " << oil;

cout << endl;

}

void Vzlet::height() {

cout << "Height==0" << endl;

}

Vzlet::~Vzlet() {

}**//Mover.h**

#pragma once

#include "Spacecraft.h"

class Mover : public Spacecraft {

public:

Mover();

void status() override;

void height() override;

~Mover();

};

Mover::Mover() {

power = "High";

oil = "Protection";

ready = "Medium";

}

void Mover::status() {

cout << "ready" << endl;

cout << "High";

cout << endl;

}

void Mover::height() {

cout << "Height==0" << endl;

}

Mover::~Mover() {

}

**//Ignition.h**

#pragma once

#include "Spacecraft.h"

class Ignition : public Spacecraft {

public:

Ignition();

void height()override;

void status()override;

~Ignition();

};

Ignition::Ignition() {

power = "Low";

oil = "Full";

ready = "Ready";

}

void Ignition::height(){

cout << "You are on the Earth" << endl;

}

void Ignition::status(){

cout << power;

cout << ready;

cout << endl;

}

Ignition::~Ignition() {

}

**//Move.h**

#pragma once

#include"Ignition.h"

#include"Mover.h"

#include"Vzlet.h"

class Move : public Ignition, public Mover, public Vzlet {

public:

Move() {

}

void status() override;

void height() override;

};

void Move::status() {

cout << "You are in the sky almost" << endl;

}

void Move::height() {

cout << "Height==2000" << endl;

}//**Polet.h**

#pragma once

#include"Move.h"

class Polet : public Vzlet, public Mover {

public:

Polet() {

}

void status() override;

void height() override;

};

void Polet::status() {

cout << "You are in the space" << endl;

}

void Polet::height() {

cout << "Height==500000" << endl;

}

**Скриншот**

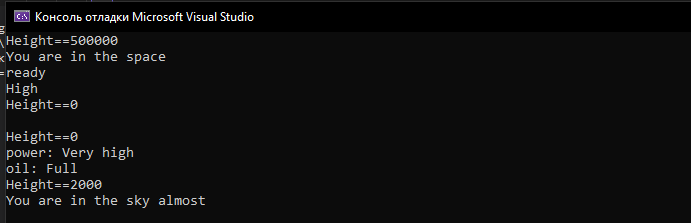
****

Рисунок 1 – результат работы программы

**Вывод**

Я изучил виды наследования и способы их применения; изучил способы использования виртуальных функций, абстрактных классов; получил навыки объектно- использования виртуальных функций, абстрактных классов; получил навыки объектно-ориентированного программирования на языке C++.