|  |  |
| --- | --- |
|  | |
| МИНОБРНАУКИ РОССИИ | |
| Федеральное государственное бюджетное образовательное учреждениевысшего образования"Московский технологический университет"МИРЭА | |
| Институт информационных технологий (ИТ) | |
| Кафедра инструментального и прикладного программного обеспечения (ИППО) | |

|  |  |
| --- | --- |
| **ОТЧЕТ ПО ЛАБОРАТОРНЫМ РАБОТАМ** | |
| **по дисциплине** | |
| **«Объектно-ориентированное программирование»** | |
|  | |
| Выполнил студент группы ИКБО-13-17 | *Хитров Н.О.* |
| Принял ассистент кафедры ИиППО | *Синицын А.В.* |

|  |  |  |
| --- | --- | --- |
| Лабораторные работы выполнены | «\_\_»\_\_\_\_\_\_\_201\_\_ г. |  |
|  |  |  |
| «Зачтено» | «\_\_»\_\_\_\_\_\_\_201\_\_ г. |  |

Москва 2018

**Оглавление**

[Лабораторная работа №6 3](#_Toc510361796)

[Цель лабораторной работы: 3](#_Toc510361797)

[Текст задания: 3](#_Toc510361798)

[UML-диаграмма: 3](#_Toc510361799)

[Код программы: 4](#_Toc510361800)

[Скриншоты выполнения программы: 6](#_Toc510361801)

[Выводы: 6](#_Toc510361802)

# Лабораторная работа №6

Цель лабораторной работы:

Целью данной лабораторной работы является изучение

наследованная классов в языке С++.

Текст задания:

1) Создать класс «Староста», производный от класса «Студент». Новый класс должен содержать несколько дополнительных методов и полей.

2) Создать класс Alive и расширить его до Bird, Fish, Animal

3) Создать класс Animal, и расширить его до Dog, Cat.

UML-диаграмма:

|  |
| --- |
| Student |
| +string name;  +string group; |
| +Student(string name, string group);  +string getName();  +void changeGroup(string newGroup); |

|  |
| --- |
| Chief: public Student |
| +bool hasJournal; |
| +Chief(bool hasJournal,string nameC, string groupC)  :Student(nameC, groupC)  +bool isJournal(); |

|  |
| --- |
| Alive |
| -isAlive; |
| +Alive(bool isAlive); |

|  |
| --- |
| Bird: public Alive |
| -bool canFly; |
| +Bird (bool isAlive, bool canFly); |

|  |
| --- |
| Fish: public Alive |
| -bool canSwim; |
| +Fish(bool isAlive, bool canSwim); |

|  |
| --- |
| Animal: public Alive |
| -int age; |
| +Animal(bool isAlive, int age);  +virtual void say() =0; |

|  |
| --- |
| Dog: public Animal |
|  |
| +void say() override; |

|  |
| --- |
| Cat: public Animal |
|  |
| +void say() override; |

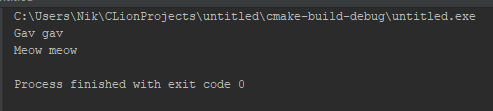
Код программы:

### Файл main.cpp:

#include <iostream>  
#include <utility>  
  
**using namespace** std;  
//#1  
**class** Student {  
**public**:  
 string name;  
 string group;  
  
 Student(string name, string group) {  
 **this**->name = std::move(name);  
 **this**->group = std::move(group);  
 }  
  
 string getName() {  
 **return** name;  
 }  
  
 **void** changeGroup(string newGroup){  
 group = std::move(newGroup);  
 }  
};  
**class** Chief: **public** Student{  
**private**:  
 **bool** hasJournal;  
**public**:  
 Chief(**bool** hasJournal,string nameC, string groupC):Student(nameC,groupC){  
 **this**->hasJournal = hasJournal;  
 }  
  
 **bool** isJournal(){  
 **return** hasJournal;  
 }  
};  
//end #1  
  
//#2  
**class** Alive{  
**private**:  
 **bool** isAlive;  
**public**:  
 **explicit** Alive(**bool** isAlive){  
 **this**->isAlive = isAlive;  
 }  
};  
  
**class** Bird: **public** Alive {  
**private**:  
 **bool** canFly;  
**public**:  
 Bird(**bool** isAlive, **bool** canFly) : Alive{isAlive} {  
 **this**->canFly = canFly;  
 }  
};  
  
**class** Fish:**public** Alive {  
**private**:  
 **bool** canSwim;  
**public**:  
 Fish(**bool** isAlive, **bool** canSwim) : Alive{isAlive} {  
 **this**->canSwim = canSwim;  
 }  
};  
  
**class** Animal: **public** Alive{  
**private**:  
 **int** age;  
**public**:  
 **explicit** Animal(**bool** isAlive, **int** age):Alive{isAlive}{  
 **this**->age=age;  
 };  
  
 **virtual void** say() =0;  
};  
//end #2

//#3  
**class** Dog: **public** Animal { ;  
**public**:  
 **explicit** Dog(**bool** isAlive, **int** age) : Animal{isAlive, age}{}  
  
 **void** say() **override** {  
 cout << "Gav gav\n";  
 }  
};  
**class** Cat: **public** Animal {  
**public**:  
 **explicit** Cat(**bool** isAlive, **int** age) : Animal{isAlive, age} {};  
  
 **void** say() **override** {  
 cout << "Meow meow\n";  
 }  
};  
//end #3  
  
**int** main() {  
 Dog dog(**true**, 5);  
 dog.say();  
 Cat cat(**true**, 7);  
 cat.say();  
 **return** 0;  
}

Скриншоты выполнения программы:



Выводы:

Изучили наследованние классов в языке С++.