

**Цель работы:** Целью данной лабораторной работы является разработка программного объекта, представляющего домашнего кота, с использованием объектно-ориентированного подхода.

**Задание:** Построить диаграмму класса для реализации описанного ниже объекта. Составить программный код и написать тестирующую программу.

Все поля класса должны быть частными (private) или защищенными (protected). Методы не должны содержать операций ввода/вывода, за исключением процедуры, единственной задачей которой является вывод информации об объекте на экран.

Объект – домашний кот. Поля: кличка, окрас и текущее состояние кота (сытый, голодный). Методы: конструктор; процедура вывода информации о коте на экран; процедура «покормить», насыщающая кота, и процедура «поиграть», заставляющая кота проголодаться.

**Выполнение:** Диаграмма класса изображена на рисунке 1

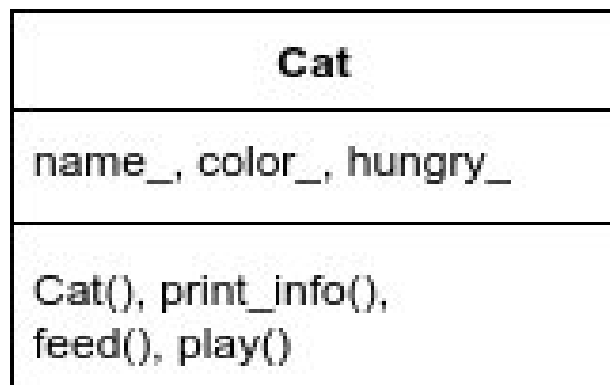


Рисунок 1 — Диаграмма класса Cat

Текст заголовочного файла и файла реализации изображены соответственно на рисунках 2 и 3

```
#ifndef CAT_H
#define CAT_H
#include <string>

using namespace std;

class Cat {
public:
    explicit Cat(const string& name, const string& color, bool hungry = true);
    void print_info();
    void feed();
    void play();

private:
    string name_, color_;
    bool hungry_;
};

#endif //CAT_H
```

Рисунок 2 — текст заголовочного файла

```
#include "Cat.h"

#include <iostream>
using namespace std;

Cat::Cat(const string& name, const string& color, const bool hungry) {
    name_ = name;
    color_ = color;
    hungry_ = hungry;
}

void Cat::print_info() {
    cout << "Cat " << name_ << ", " << color_ << ", " << (hungry_? "hungry." : "well-fed.") << endl;
}

void Cat::feed() {
    hungry_ = false;
}

void Cat::play() {
    hungry_ = true;
}
```

Рисунок 3 — текст файла реализации

Текст основной (тестирующей) программы изображен на рисунке 4

```
#include "Cat.h"

#include <iostream>
using namespace std;

int main() {
    Cat cat1{name: "Murzik", color: "Black", hungry:false};
    cat1.print_info();
    cat1.play();
    cat1.print_info();
    cat1.feed();
    cat1.print_info();

    cout << "-----" << endl;

    Cat cat2{name: "Funtik", color: "Gray", hungry:true};
    cat2.print_info();
    cat2.feed();
    cat2.print_info();
    cat2.play();
    cat2.print_info();

    return 0;
}
```

Рисунок 4 — текст основной (тестирующей) программы

**Результат выполнения программы:** результат выполнения программы, описанной выше изображен на рисунке 5

```
Cat Murzik, Black, well-fed.  
Cat Murzik, Black, hungry.  
Cat Murzik, Black, well-fed.  
-----  
Cat Funtik, Gray, hungry.  
Cat Funtik, Gray, well-fed.  
Cat Funtik, Gray, hungry.
```

Рисунок 5 — результат выполнения программы

**Вывод:** В процессе выполнения лабораторной работы был разработан и реализован класс, представляющий объект домашнего кота. С помощью методов класса были продемонстрированы базовые операции, такие как вывод информации об объекте, кормление кота и игра с ним. Класс обеспечил правильную инкапсуляцию данных, так как все поля были объявлены как частные, а доступ к ним осуществлялся через методы.