

**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ЯДЕРНЫЙ УНИВЕРСИТЕТ
«МИФИ»**

Институт Лаплас
Кафедра №31 «Прикладная математика»

ЛАБОРАТОРНАЯ РАБОТА №4
по дисциплине
«Объектно-ориентированное программирование»

Задание

Написать программу на C++, которая при создании единственного объекта класса `One` и выхода из программы

```
int main()
{
    One one;
}
```

выводит на экран строго определенную последовательность слов:

One Two Three Four Five Six Seven Eight Nine Ten.

Центральным элементом задачи является класс `One`. Его определение (указано для каждого варианта) изменять нельзя, за исключением текста внутри вывода. `[WORD]` можно заменить любым словом (не обязательно одним и тем же). Реализовать нужно только следующие классы: `Two`, `Three`, `Four`, `Five`. Каждый созданный вами класс не может выводить больше одного слова в своём конструкторе и одного слова в своём деструкторе.

Структура класса `One` содержит все необходимые подсказки. Чтобы успешно решить задачу обратите внимание на следующие моменты:

1. Поля-ссылки внутри класса не создают новый экземпляр класса, а должны быть связаны с уже существующим объектом, совместимым по типу.
2. При наличии поля-указателя следует помнить, что объект, на который ссылается указатель, не создается и не уничтожается автоматически. Его время жизни полностью контролируется командами `new` и `delete`.

Выбор варианта

Номер варианта определяется по формуле: $n = (k \bmod 4) + 1$, где k — номер студента в списке группы.

Вариант 1

```
class One : public Two
{
public:
    One() : five_ref(*this)
    {
        Three three;
        four_ptr = new Four();
        cout << "[WORD] ";
    }
    ~One()
    {
        delete four_ptr;
        cout << "[WORD] ";
    }
private:
    Four* four_ptr;
    Five& five_ref;
};
```

Вариант 2

```
class One : public Two
{
public:
    One() : five_ptr(this), three_ref(four)
    {
        cout << "[WORD] ";
    }
    ~One()
    {
        cout << "[WORD] ";
    }
private:
    Four four;
    Five* five_ptr;
    Three& three_ref;
};
```

Вариант 3

```
class One : public Two
{
public:
    One() : four(five)
    {
        cout << "[WORD] ";
    }
    ~One()
    {
        cout << "[WORD] ";
    }
private:
    using Three::five;
    Four four;
};
```

Вариант 4

```
class One : public Two
{
public:
    One() : five(*four_ptr)
    {
        cout << "[WORD] ";
    }
    ~One()
    {
        cout << "[WORD] ";
    }
private:
    using Three::four_ptr;
    Five five;
};
```