

## Практическая работа: Пространства имен и выделение памяти (new/delete)

---

### Задача 1: Пространство имен для математических операций

Создайте пространство имен MathUtils, содержащее:

- Функцию `square(x)`, возвращающую квадрат числа.
  - Константу `PI = 3.1415`.
- В `main()` выведите результат `square(5)` и значение `PI`, используя оператор `::`.
- 

### Задача 2: Вложенные пространства имен

Объявите пространство имен `Game`, внутри него — `Physics`, содержащее функцию `applyGravity()`, которая выводит "Gravity applied!".  
В `main()` вызовите `applyGravity()` через вложенные пространства.

---

### Задача 3: Псевдонимы для пространств

Создайте псевдоним `Short` для пространства имен `VeryLongNamespaceName`.  
Пример:

```
namespace VeryLongNamespaceName { void test() {} }  
// Ваш код для псевдонима  
Short::test(); // Должно работать
```

---

### Задача 4: Исправьте ошибку (using namespace)

Почему этот код не компилируется? Исправьте:

```
namespace A { int x = 10; }  
namespace B { int x = 20; }  
using namespace A;  
using namespace B;  
int main() { std::cout << x; }
```

---

### Задача 5: Динамическое создание объекта

Напишите код:

1. Создайте класс `Student` с полем `name` (строка) и конструктором.
2. Выделите память для объекта `Student` с именем "Alice" через `new`.

3. Удалите объект через delete.

---

### Задача 6: Массивы и new/delete

Создайте динамический массив из 10 целых чисел. Заполните его значениями от 1 до 10. Выведите массив и освободите память.

---

### Задача 7: Обработка исключений

Напишите код, который пытается выделить память для массива из  $10^{12}$  элементов типа int. Обработайте исключение std::bad\_alloc, выводя "Out of memory!".

---

### Задача 8: Поиск утечек

Найдите и исправьте 3 ошибки в коде:

```
int* a = new int(5);
int* arr = new int[3]{1, 2, 3};
delete a;
```

---

### Задача 9: Умные указатели

Перепишите код из задачи 5, используя std::unique\_ptr. Объясните, почему это лучше.

---

### Задача 10: Комплексная задача

Создайте:

1. Пространство имен Library с классом Book (поле title).
2. Динамически создайте массив из 3 объектов Book с разными названиями.
3. Используйте std::shared\_ptr для управления памятью.
4. Выведите названия книг и убедитесь, что память освобождается автоматически.

---

### Пример решения для задачи 1:

```
#include <iostream>

namespace MathUtils {
```

```
const double PI = 3.1415;
int square(int x) { return x * x; }
}

int main() {
    std::cout << "Square: " << MathUtils::square(5) << std::endl;
    std::cout << "PI: " << MathUtils::PI << std::endl;
    return 0;
}
```

---