

Программирование на C++



| Минцифры
РОССИИ

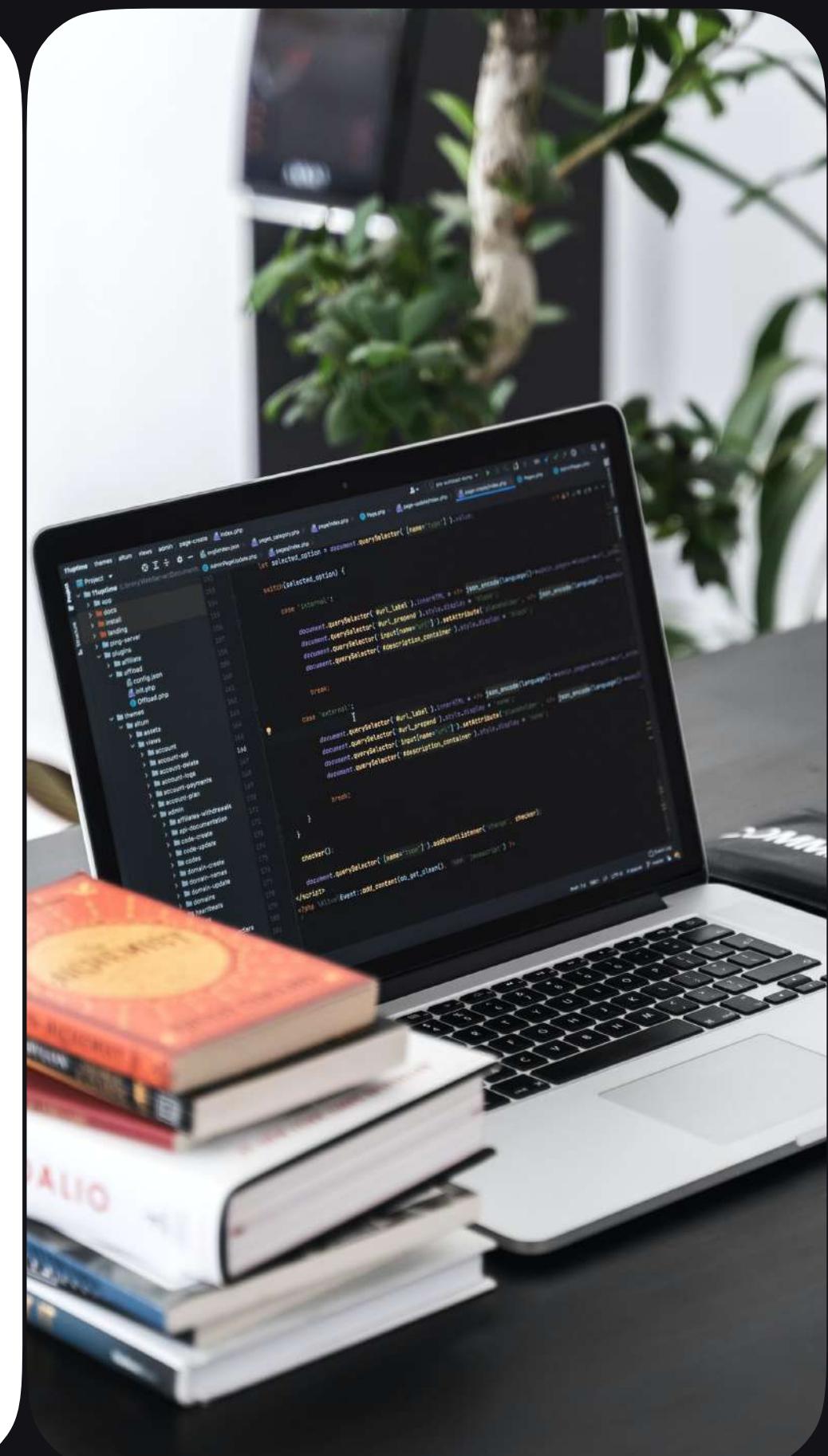
UCHI DOMA

20.35
УНИВЕРСИТЕТ

Урок 5 Модуль 3

Создание собственных классов

Полезные материалы



Цели урока

- ❖ изучить конструкторы и деструкторы
- ❖ отработать на практике написание алгоритмов с конструкторами и деструкторами на C++



Конструктор

Конструктор – функция, предназначенная для инициализации объектов класса

ООП дает возможность программисту описать функцию, явно предназначенную для инициализации объектов. Поскольку такая функция конструирует значения данного типа, она называется **конструктором**. Конструктор всегда имеет то же имя, что и сам класс и никогда не имеет возвращаемого значения.

Когда класс имеет конструктор, все объекты этого класса будут проинициализированы.

Конструктор

```
class date {  
    int day, month, year;  
public:  
    date(int, int, int); // конструктор  
};
```

Конструктор

Если конструктор требует аргументы, их следует указать:

```
date today = date(10,01,2023); // полная форма
```

```
date xmas(10,01,2023); // сокращенная форма
```

Неверное создание объекта.

```
date my_burthday; // недопустимо, опущена инициализация
```

Конструктор

Если необходимо обеспечить несколько способов инициализации объектов класса, задается несколько конструкторов:

```
class date {  
    int month, day, year;  
  
public:  
    date(int, int, int); // день месяц год  
    date(char*); // дата в строковом представлении  
    date(); // дата по умолчанию: сегодня  
};
```

Конструктор

Если конструкторы существенно различаются по типам своих параметров, то компилятор при каждом использовании может выбрать правильный:

```
date july4("Февраль 27, 2014");  
date guy(27, 2, 2014);  
date now; // инициализируется по умолчанию
```

Конструктор по умолчанию

Конструктор, не требующий параметров, называется конструктором по умолчанию.

```
class date {  
    int month, day, year;  
public:  
    date(int, int, int); // день месяц год  
    date(char*); // дата в строковом представлении  
    date(); // дата по умолчанию: сегодня  
};
```

Деструкторы



Определяемый пользователем класс имеет конструктор, который обеспечивает надлежащую инициализацию. Для многих типов также требуется обратное действие. Деструктор обеспечивает соответствующую очистку объектов указанного типа.



Имя деструктора представляет собой имя класса с предшествующим ему знаком «тильда» ~.



Так, для класса X деструктор будет иметь имя ~X().



Многие классы используют динамическую память, которая выделяется конструктором, а освобождается деструктором.

Деструкторы

```
class date
{
    int day, year;
    char *month;

public:
    date(int d, char* m, int y)
    {
        day = d;
        month = new char[strlen(m)+1]; //выделяем память
        strcpy_s(month, strlen(m)+1, m); //копируем строку
        year = y;
    }
    ~date() { delete[] month; } // деструктор
};
```