



9-11 классы

## Программирование на C++

Презентация занятия

# Структурные типы данных

10 занятие



Минцифры  
России



**20.35**  
УНИВЕРСИТЕТ

Программирование  
на C++

# Теоретическая часть

Структурные типы данных

10 занятие



**инжинириум®**

МГТУ им. Н.Э. Баумана

2020



## Тема: Структурные типы данных

```
struct book {  
    char title[50];  
    char author[30];  
    int pages;  
};  
struct circle {  
    int x, y;  
    float dia;  
    char color[10];  
};
```

```
struct circle a, b, c;  
struct book mybook;
```





## Тема: Структурные типы данных

```
a.x = 10; a.dia = 2.35;  
printf("%.2f ", a.dia);
```

```
struct book lang_c = { "Language C", "Ritchi", 99 };
```



## Тема: Структурные типы данных

Значение переменной-структуры можно присвоить переменной того же типа:

```
struct book {  
    char* title,  
        * author;  
    int pages;  
};  
struct book old, new;  
old.title = "GNU/Linux";  
old.author = "people";  
old.pages = 20213;  
new = old;  
new.pages += 2000;  
printf("%d, %d\n", old.pages, new.pages);
```



## Тема: Структурные типы данных

Структуры-переменные можно передавать в функции в качестве параметров и возвращать их оттуда. Структуры передаются по значению, как обычные переменные, а не по ссылке, как массивы.

Рассмотрим программу, в которой одна функция возвращает структуру, а другая — принимает ее в качестве параметра





## Тема: Структурные типы данных

```
#include <stdio.h>
```

```
#include <math.h>
```

```
struct circle {  
    int x, y;  
    float dia;  
    char color[10];  
};
```





## Тема: Структурные типы данных

```
#include <stdio.h>
```

```
#include <math.h>
```

```
struct circle {  
    int x, y;  
    float dia;  
    char color[10];  
};  
struct circle new_circle();  
void cross(struct circle);  
int main() {  
    struct circle a;  
  
    a = new_circle();  
    cross(a);  
}
```







## Тема: Структурные типы данных

```
struct circle new_circle() {  
    struct circle new;  
  
    printf("Координаты: ");  
    scanf("%d%d", &new.x, &new.y);  
    printf("Диаметр: ");  
    scanf("%f", &new.dia);  
    printf("Цвет: ");  
    scanf("%s", new.color);  
  
    return new;  
}
```





## Тема: Структурные типы данных

```
void cross(struct circle c) {  
    double hyp;  
  
    hyp = sqrt((double)c.x * c.x +  
               (double)c.y * c.y);  
    printf("Расстояние: %.2f\n", hyp);  
    if (hyp <= c.dia / 2)  
        puts("Пересекает");  
    else  
        puts("Не пересекает");  
}
```



## Тема: Структурные типы данных

```
// переменная-структура  
struct book new;
```

```
// указатель на структуру  
struct book* pnew;
```

```
// передаем адрес  
reader(&new);
```

```
void reader(struct book* pb);
```

```
pnew = &new;
```

```
// передаем указатель  
reader(pnew);
```

Программирование  
на C++

# Практическая часть

Структурные типы данных

10 занятие



2020

## Тема: Статическая и динамическая память

### Задание 1

1. Описать структуру с именем AEROFLOT, содержащую следующие поля:
  - название пункта назначения рейса;
  - номер рейса;
  - тип самолета.
2. Написать программу, выполняющую следующие действия:
  - ввод с клавиатуры данных в массив, состоящий из семи элементов типа AEROFLOT; записи должны быть размещены в алфавитном порядке по названиям пунктов назначения;
  - вывод на экран пунктов назначения и номеров рейсов, обслуживаемых самолетом, тип которого введен с клавиатуры;
  - если таких рейсов нет, выдать на дисплей соответствующее сообщение

## Тема: Статическая и динамическая память

```
#include <iostream>
#include <cstring>
#define M 15
using namespace std;

struct AEROFLOT {
    char kuda[M];
    int num;
    char type[M];
};
```

## Тема: Статическая и динамическая память

```
int main() {  
    static int N = 3;  
    AEROFLOT fly[N];  
    int i;  
    for (i = 0; i < N; i++) {  
        cout << "\nВведите номер рейса: "; cin >> fly[i].num;  
        cout << "\nВведите тип самолёта "; cin >> fly[i].type;  
        cout << "\nВведите пункт назначения "; cin >> fly[i].kuda;  
    }
```



## Тема: Статическая и динамическая память

```
for (i = 0; i < N - 1; i++)  
    for (int j = i + 1; j < N; j++)  
        if (strcmp(fly[i].kuda, fly[j].kuda) > 0)  
        {  
            strcpy(cur, fly[i].kuda);  
            strcpy(fly[i].kuda, fly[j].kuda);  
            strcpy(fly[j].kuda, cur);  
        }
```







## Тема: Статическая и динамическая память

```
cout << "\n\n";  
for (i = 0; i < N; i++) {  
    cout << fly[i].kuda << " " << fly[i].num << " " << fly[i].type << endl;  
}  
cout << "\n\n";
```



## Тема: Статическая и динамическая память

```
char temp[M];
int count;
while (strcmp(temp, "exit")) {
    cout << "\nВведите желаемый тип самолёта: "; cin >> temp;
    count = 0;
    for (i = 0; i < N; i++) {
        if (strcmp(fly[i].type, temp) == 0) {
            cout << endl << fly[i].kuda << "    " << fly[i].num << endl;
            count++;
        }
    }
    if (count == 0) cout << endl << " Нет таких рейсов\n";
}
```