

Семинар №7

ФАКИ 2016

Бирюков В. А.

October 28, 2016

- `#include` — вставляет текст из указанного файла
- `#define` — задаёт макрос или символическую константу
- `#if`
- `#ifdef`
- `#endif`
- `#pragma`

Typedef

- Typedef – ключевое слово в языке C
- Используется для того, чтобы дать типу новое имя

- `typedef int my_new_int;`

```
typedef unsigned long long ull;
```

• Структуры

Структуры

- Структура – это композитный тип данных, группирующий, без сокрытия набор значений

- ```
struct tag_name {
 type1 member1;
 type2 member2;
 /* ... */
};
```

# Структуры

## Пример

Описание структуры:

```
struct account {
 int account_number;
 char first_name[30];
 char last_name[50];
 float balance;
};
```

Объявление структуры:

```
struct account ac1;
```

# Структуры

## Пример

Инициализация структуры:

```
struct account ac1 = {1, "Ivan",
 "Ivanov", 1000.0};
```

Доступ к элементу структуры(оператор .):

```
ac1.first_name = "Petr";
ac1.balance += 100;
printf("%s has %.2f roubles\n",
 ac1.last_name, ac1.balance);
```

# Typedef

- Typedef – ключевое слово в языке C
- Используется для того, чтобы дать типу новое имя
- `typedef unsigned long long ull;`

```
typedef struct {
 int account_number;
 char first_name[30];
 char last_name[50];
 float balance;
} account;
```



# Структуры и функции

## Передача структур по значению

```
void print(account ac)
{
 printf("%s has %.2f roubles\n",
 ac.last_name, ac.balance);
}
```

# Структуры и функции

Передача структур с помощью указателей

```
void print(account * ac)
{
 printf("%s has %.2f roubles\n",
 ac->last_name, ac->balance);
}
```

# Структуры и функции

Передача и возврат структур по значению

```
account add_salary(account ac)
{
 ac.balance += 30000.0;
}
```

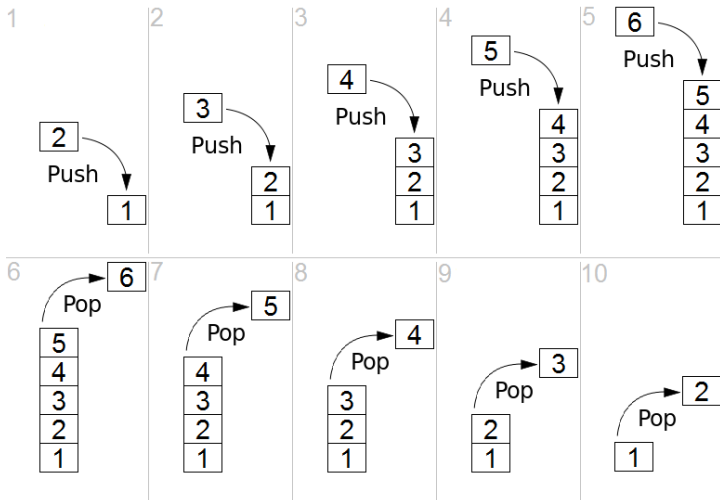
# Структуры и функции

Передача и возврат структур с помощью указателей

```
void add_salary(account * ac)
{
 ac->balance += 30000.0;
}
```

- Стек и очередь

## Стек



# Стек

## Описание стека

```
#define N 100
typedef int Data;

struct Stack {
 int n;
 Data a[N];
};
```

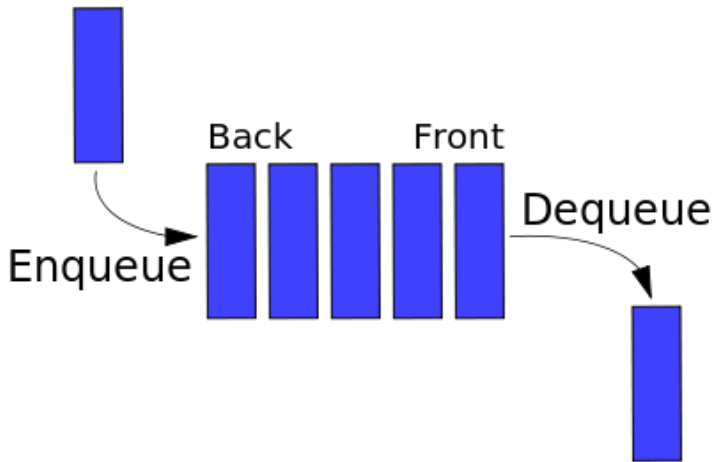
# Стек

## Добавление элемента в стек

```
void stack_push(struct Stack * s, Data
 x)
{
 s->Data[s->n] = x;
 s->n++;
}
```



# Очередь



# Задание

# Задание

- Задачи на структуры: контекст Структуры
- Задачи на стек: контекст Преобразователь стека в очередь