

# Информатика (основы программирования)

Лекция 7.

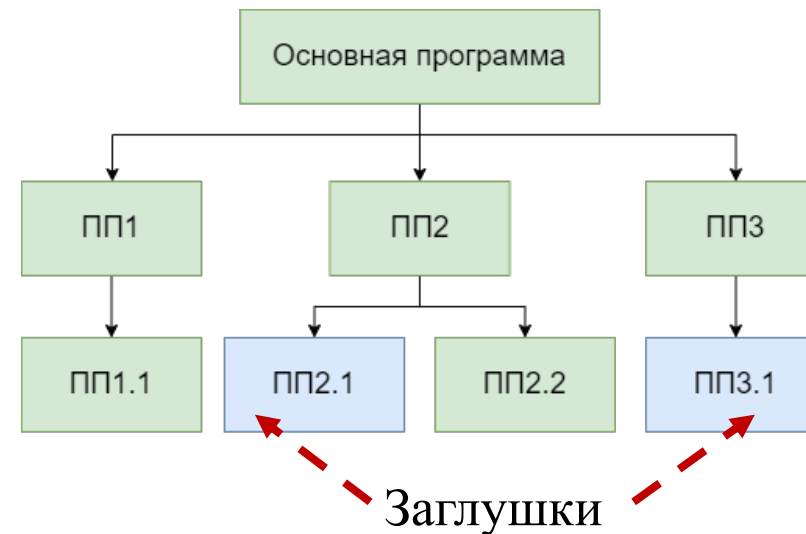
Структурное программирование на языке C

Автор: Бабалова И.Ф.

Доцент, каф.12

# Структурный подход к проектированию программ

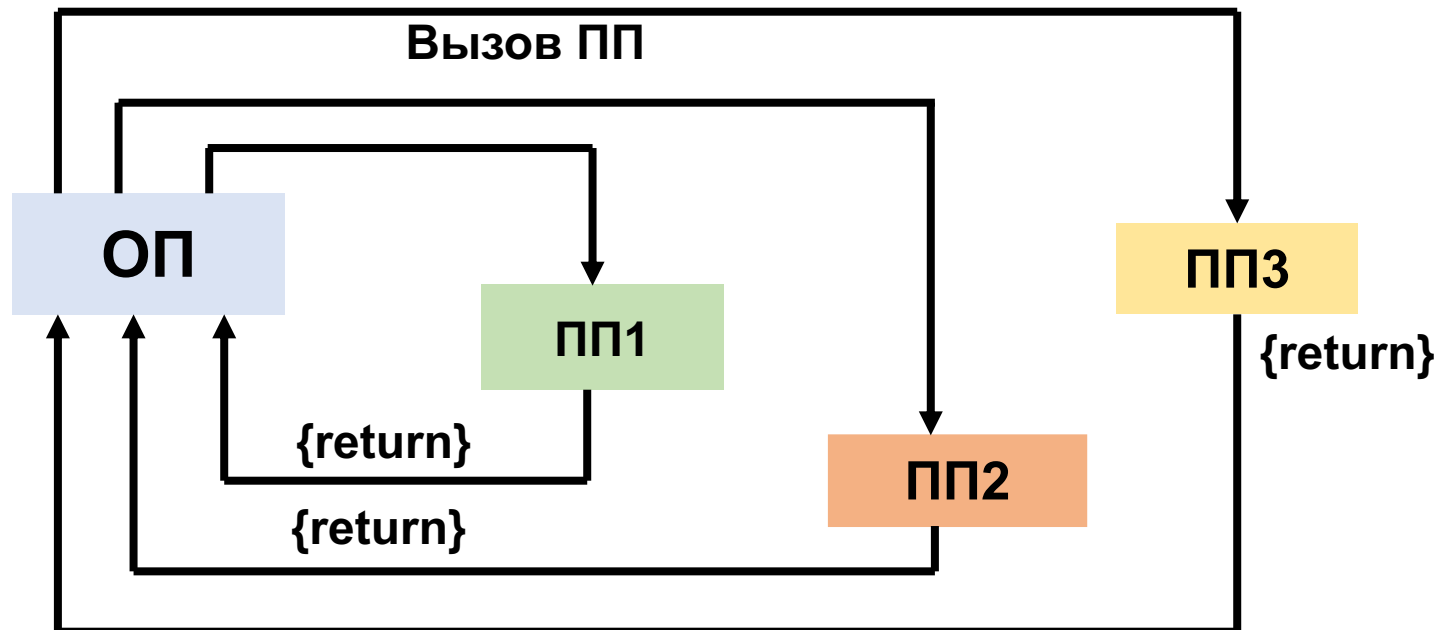
- Структурное проектирование программ позволяет создавать программу из отдельных хорошо структурированных компонент.
- Проектирование программы может выполняться сверху вниз (нисходящее проектирование) или снизу вверх (восходящее проектирование).
- Пример нисходящего проектирования:



- Контроль правильности работы программы в каждой ветви блок-схемы может выполняться независимо от возможностей решения задачи по другим ветвям.

# Подпрограммы (1)

- Подпрограмма (ПП) - часть программы, которая имеет свое уникальное имя и может быть выполнена многократно с разными значениями параметров.
- С помощью ПП удастся структурировать текст программы. Каждая ПП позволяет реализовать некоторый набор действий или получить конкретный результат.



# Подпрограммы (2)

- Основная программа
  - Объявления глобальных типов и переменных
  - Описание функций
  - Объявления локальных типов данных и переменных
- Действие только глобальных объявлений

ТИПОВ



# Подпрограммы (3)

Требования к списку формальных параметров подпрограмм:

1. Передаются в подпрограмму те параметры, которые необходимы для её выполнения.
  2. Глобальные переменные могут передаваться в подпрограмму. Но в этом случае они могут быть изменены за её пределами.
  3. В списке формальных параметров может присутствовать параметр результата.
  4. В списке формальных параметров не нужны индексы массивов.
- Индексы массивов — это локальные переменные для всех подпрограмм.
  - Различия в локальных и глобальных параметрах!

# Функции (1)

- Подпрограммы введены во все языки программирования на основании принципа декомпозиции алгоритмов, что предполагает разбиение большой задачи на подзадачи. Необходимость такого разбиения обоснована в теории алгоритмов.
- В языке С подпрограммы реализованы в виде функций.

# Функции (2)

- **Функция** — это совокупность объявлений и операторов, обычно предназначенная для решения определенной задачи.
- Каждая функция должна иметь **имя**, которое используется для ее объявления и вызова. В любой программе на языке C должна быть функция с именем `main` (главная функция). Именно с этой функции, в каком бы месте программы она не находилась, начинается выполнение программы.

# Функции (3)

- **Функция** — структурная единица программы, которая в основную программу возвращает некоторое значение.
- Описание функций включает в себя её объявление (заголовок функции) и её определение (заголовок + тело функции).
- Заголовок (прототип) функции:  
[<тип результата или void>] <имя>  
(<перечень формальных параметров>)
- Определение функции:  
[<тип результата или void>] <имя>  
<(перечень формальных параметров)>  
<тело функции (объявления, операторы и т.д.)>



# Функции (4)

- Компилятор языка C является однопроходным (обрабатывает каждый файл один раз).
- Заголовок функции (а возможно – и её определение) должен быть описан перед её вызовом. Например, для того, чтобы вызвать функцию `f1()` из функции `main()`, заголовок функции `f1()` должен быть описан до определения функции `main()`.
- **Нельзя описать функцию в теле функции!**

# Пример определения и использования функций

## Файл operations.h

```
#ifndef OPERATIONS_H
#define OPERATIONS_H
/* Получение случайных чисел на */
/* заданном отрезке [x,y] */
int srd(int x, int y);
#endif
```

## Файл operations.c

```
#include <stdlib.h>
#include "operations.h"
int srd(int x, int y) {
    int q = x + rand() % (y - x + 1);
    return q;
}
```

## Файл main.c

```
#include <stdio.h>
#include <stdlib.h>
#include "operations.h"
int main() {
    srand(time(NULL));
    ...
    for (int i = 0; i < len; ++i) {
        a[i] = srd(c, b);
        printf("%d ", a[i]);
    }
    printf("\n");
    ...
}
```

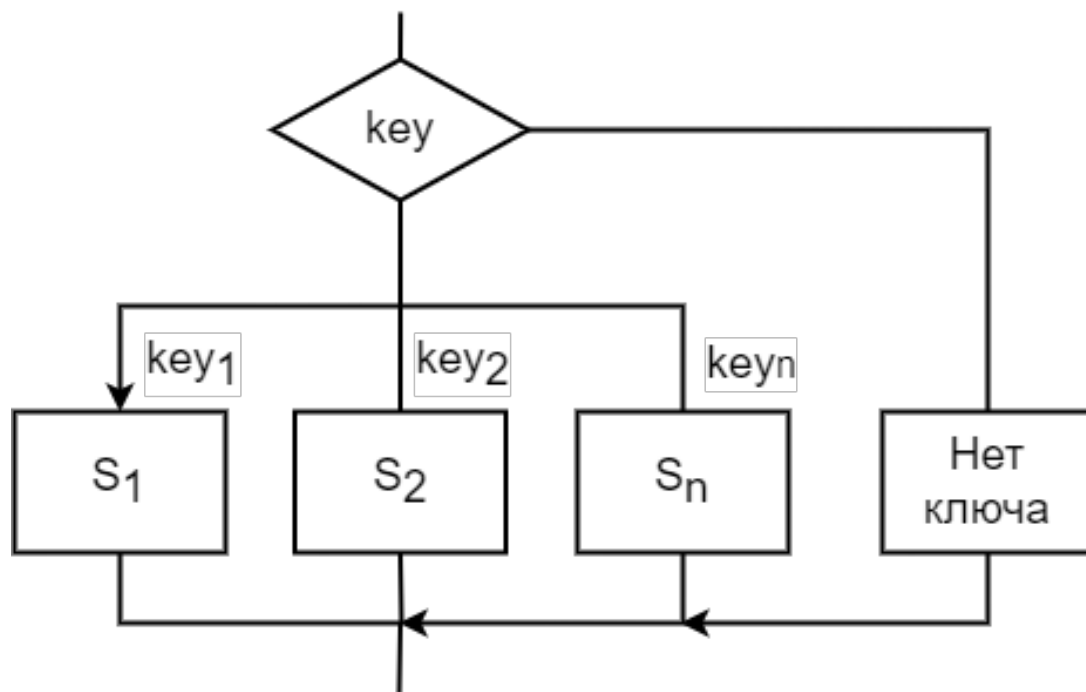
**Сборка программы: `cc -o program operations.c main.c`**

# Оператор switch (1)

Оператор **switch** предназначен для организации выбора решения из множества различных вариантов. Формат оператора следующий:

```
switch (выражение) {  
    <объявление переменных блока>  
    case <константа1>: <список-операторов1>  
    case <константа2>: <список-операторов2>  
    default: <список операторов>  
}
```

# Оператор switch (2)



```
int i = 2;
switch (i) {
    case 1:
        i += 2;
    case 2:
        i *= 3;
    case 0:
        i /= 2;
        break;
    case 4:
        i -= 5;
    default;
}
```

# Пример текстового меню

```
int main() {
    char option = -1;
    do {
        printf("\n");
        printf("***** MENU *****\n");
        printf("1. Option 1\n");
        printf("2. Option 2\n");
        printf("0. Exit\n");
        printf("\n");
        if (scanf(" %c", &option) == EOF) {
            // Ctrl + D
            return 0;
        }
        switch (option) {
            case '1':
                func1(...);
                break;
            case '2':
                func2(...);
                break;
            default:
                printf("Wrong input!\n");
                break;
        }
    } while (option != '0');
    return 0;
}
```

Thank you