

# Лекция 3

Алгоритмические языки и  
программирование

# Часть 1

# ФУНКЦИИ

- Функции - это базовые блоки Си, в которых выполняются разные операции.
- Стандартный вид функций следующий:

*тип-результата имя-функции (список параметров)*

```
{  
    объявления;  
    инструкции;  
    тип-результата;  
}
```

# Оператор выбора switch

- Оператор выбора **switch** является очень удобной заменой множественного использования операторов **if**.
- Оператор **switch** сравнивает значение одной переменной с несколькими константами.
- Основной формат для использования оператора множественного выбора **switch case** показан — ниже.
- Значение переменной указанной в условии **switch** сравнивается со значениями, которые следуют за ключевым словом **case**. Когда значение в переменной, соответствует значению в строке с оператором **case**, компьютер продолжит выполнение программы с этого

места.

```
switch (<переменная>) {  
    case <значение 1>:  
        <ветвь 1>;  
        break;  
    case <значение 2>:  
        <ветвь 2>;  
        break;  
    default:  
        <действия по умолчанию>;  
break;  
}
```

# ФУНКЦИИ

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

int multiplication( int num1, int num2 ); //прототип функции

int main()
{
    int num1;
    int num2;
    setlocale (LC_All, "rus");
    printf("Введите два числа для умножения: ");
    scanf("%d", &num1);
    scanf("%d", &num2);
    printf("Результат умножения %d\n", multiplication(num1, num2)); // ВЫЗОВ функции
    return 0;
}

int multiplication(int num1, int num2) // определение функции
{
    return num1 * num2;
}
```

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

# Оператор выбора switch

```
int main()
{
    int input;
    printf( "Что бы сегодня посмотреть?\n" );
    printf( "1. Стрела(Arrow)\n" );
    printf( "2. Сверхъестественное(Supernatural)\n" );
    printf( "3. Ходячие мертвецы(Walking Dead)\n" );
    printf( "4. Выход\n" );
    printf( "Ваш выбор: " );
    scanf( "%d", &input );
    switch ( input ) {
        case 1:                /* обратите внимание на двоеточие, точки сзапятой тут нет */
            printf("«His Death Was Just The Beginning»\n");
            break;
        case 2:
            printf("«Scary»\n");
            break;
        case 3:
            printf("«Fight the dead. Fear the living»\n");
            break;
        case 4:
            printf( "Сегодня смотреть ничего не будем :(\n" );
            break;
        default:
            printf( "Неправильный ввод.\n" );
    }
    getchar();
    return 0;
}
```

# Часть 2

# Явное и неявное преобразование ТИПОВ ДАННЫХ

- В Си различают **явное** и **неявное** преобразование типов данных.
- Неявное преобразование типов данных выполняет компилятор Си.
- Явное преобразование данных выполняет сам программист.
- Результат любого вычисления будет преобразовываться к наиболее точному типу данных, из тех типов данных, которые участвуют в вычислении.



# Неявное приведение типов

х	у	Результат деления	Пример
делимое	делитель	частное	$x = 15 \ y = 2$
int	int	int	$15/2=7$
int	float	float	$15/2=7.5$
float	int	float	$15/2=7.5$

# Явное приведение типов

- `float(15) / 2 ; // результат равен 7.5, число 15 преобразуется в вещественный тип данных float.`
- `double(15) / 2; // результат равен 7.5 – тоже самое!!!`

# Генератор случайных чисел rand() в Си

- Случайные числа в Си могут быть сгенерированы функцией rand() из стандартной библиотеки Си.
- Функция rand() генерирует числа в диапазоне от 0 до RAND\_MAX.
- RAND\_MAX — это константа, определённая в библиотеке <stdlib.h>.
- Для RAND\_MAX = 32767, но оно может быть и больше, в зависимости от компилятора.

# Генератор случайных чисел rand() в Си

```
// формула генерации случайных чисел по заданному диапазону  
int r;  
r = firs_value + rand() % last_value;  
// где firs_value - минимальное число из желаемого диапазона  
// last_value - ширина выборки
```

# Генератор случайных чисел rand() в Си

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

int main() {
    int i, r;

    srand(time(NULL));
    for (i = 0; i < 10; i++) {
        r = rand()%3+1;
        printf("%d\n", r);
    }
    return 0;
}
```

# Часть 2

# Рекурсивный вызов функций

- **Рекурсия** — вызов функции из неё же самой, непосредственно (*простая рекурсия*) или через другие функции (*сложная или косвенная рекурсия*), например, функция {A} вызывает функцию {B}, а функция {B} — функцию {A}. Количество вложенных вызовов функции называется глубиной рекурсии.

# Рекурсивный вызов функций

```
#include <stdio.h>
/* printf: печатает n как целое десятичное число */
void printd(int n);

int main(){
    int n = 153;
    printd(n);
    getchar();
    return 0;
}

void printd(int n){
    if (n < 0) {
        printf("-");
        n = -n;
    }
    if (n / 10)
        printd(n / 10);
    printf("%d", n % 10);
}
```



# Лабораторные работы

# Возведение в степень

- Напишите программу, которая будет возводить в степень числа в заданном диапазоне.
- Пример:

Введите диапазон от a до b и основание степени base:

a=3

b=5

base=2

Результат возведения в степень: 8, 16, 32

- Примечание:
  1. Использовать циклы
  2. Использовать функции

# Пальцы

- Пользователь вводит порядковый номер пальца руки. Напишите программу, которая показывает название пальца на экране.

# Псевдослучайные числа

- Напишите программу, которая в соответствии с заданными числами 1, или 2, или 3 будет генерировать пять псевдослучайных чисел в заданном диапазоне и записывать их в массив. Массив выводить на экран.

Выберите режим работы программы:

- 1) Генерация числа от 1 до 5
- 2) Генерация числа от 5 до 10
- 3) Генерация числа от 5 до 100000

>2

Вывод: 5, 7, 6, 9, 8

- **Примечание:**

1. Использовать `switch()`;
2. Использовать массивы
3. Использовать функции `srand()`; и `rand()`;

Дополнительно:

1. Использовать функции

# Поиск

- Заполните массив случайными числами. Пользователь вводит 2 числа, найти их в сгенерированном массиве.
- Вывести эти числа из массива.

# Калькулятор

- Напишите программу, которая реализует калькулятор из 6 математических операций.

- Примечание:

1. Использовать `switch()`, для выбора операций;
2. Использовать функции для каждой операции;
3. Использовать приведение типов, где необходимо;

Символ операции	Название операции
*	умножение
/	деление
+	сложение
-	вычитание
<code>pow(a,b)</code>	возведение в степень
<code>sqrt(x)</code>	вычисление корня

Выберите режим работы программы:

- 1) Сложение
- 2) Вычитание
- 3) Деление

. . .

>2

Введите два числа:

>2

>1

Ответ: 2

# Числа

- Дано натуральное число  $n$ . Напишите программу, которая выведет все числа от 1 до  $n$ .
  - Примечание:
    1. Использовать рекурсивную функцию;

# Факториал

- Напишите программу, вычисляющую факториал.
  - Примечание:
    1. Использовать рекурсивную функцию;