

# Основы языка C

Семестр 1

Семинар 1

# Организационное

- Беклемышева Катерина Алексеевна  
[amisto@yandex.ru](mailto:amisto@yandex.ru)
- Студенты
  - ФИО
  - Факультетская группа
  - e-mail

# Курсы

1. Основы информатики (С)
2. Архитектура ЭВМ и язык ассемблера
3. Операционные системы (UNIX)
4. ООП (С++)

# Литература

- Ворожцов А. В., Винокуров Н. А. Практика и теория программирования. – М.: Физматкнига, 2008.
- Керниган, Ритчи  
"Язык программирования С»
- книги из рассылки

# IDE для C/C++

Integrated Development Environment

## Windows

- DevCpp
- CodeBlocks
  - CLion
- Microsoft Visual Studio

## Linux

Fedora, Ubuntu...

- vim/gedit + gdb
- CodeBlocks
  - CLion
  - Geany

# Hello, world

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

Включение информации о стандартной библиотеке ввода-вывода  
(без этой строки printf не работает)

```
int main ( )  
{
```

С функции **main** начинается выполнение программы

```
printf ("Hello, world!\n");
```

Печать строки

```
return 0;
```

```
}
```

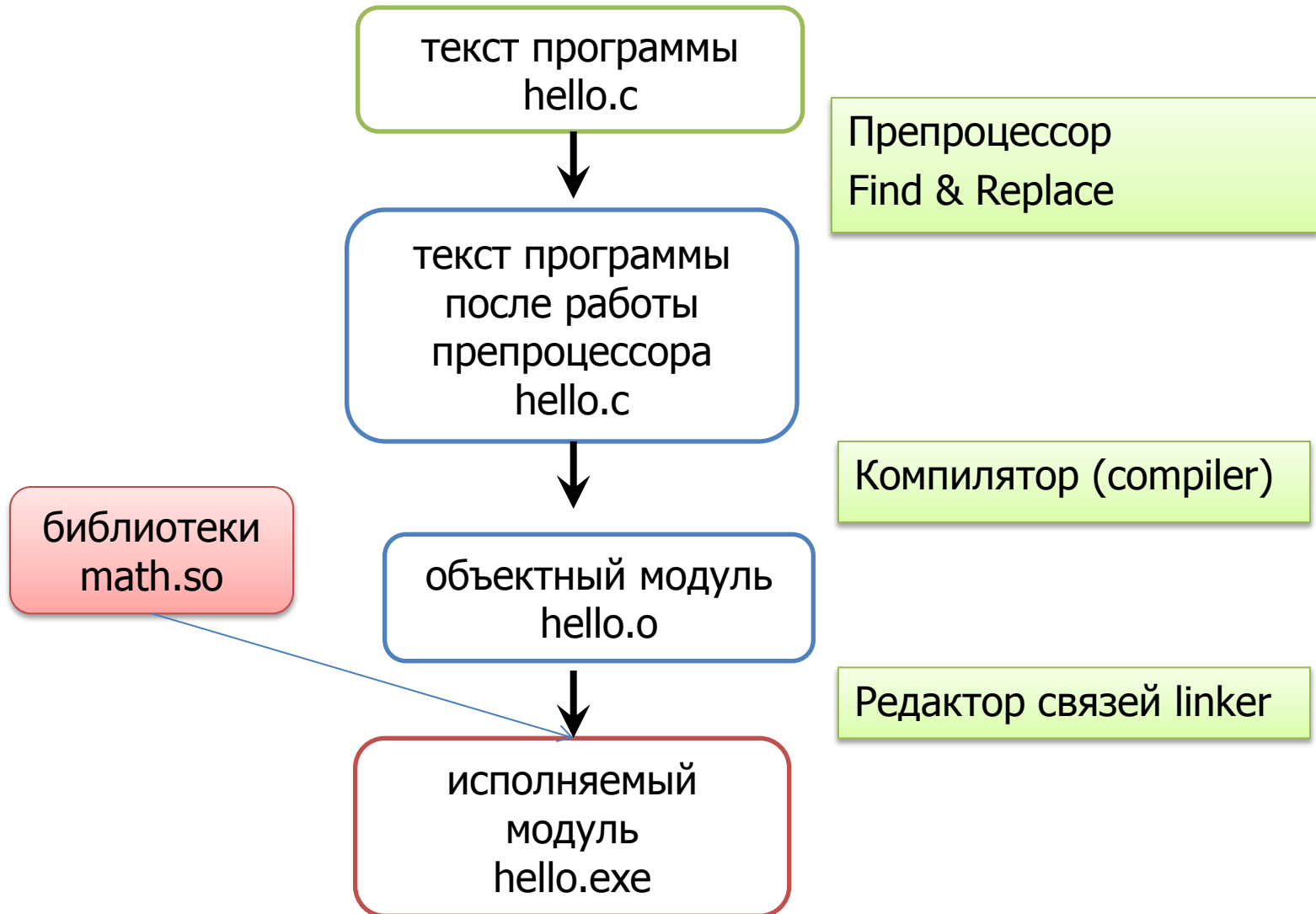
Возвращаем целое число (обычно код ошибки)

# Escape - последовательности

## Экранирующие последовательности

- `\n` – new line
- `\r` – возврат каретки
- `\t` – табуляция
- `\b` – backspace
- `\"` – кавычки
- `\\` – \
- `\a` – alarm (звуковой сигнал)

# От текста к бинарному коду





# Задание

- Запустить DevC++ / CodeBlocks / Visual Studio
- Набрать текст программы Hello, world
- Сохранить файл
- Скомпилировать программу
- Запустить программу
- Научить соседа

# Из цельсия в фаренгейты

```
#include <stdio.h>
int main ( )
{
    int fahr, cels, low, up, step;
    low = 0;
    up = 300;
    step = 20;
    fahr = low;

    while (fahr <= up) {
        cels = 5 * (fahr - 32) / 9;
        printf ("%d\t%d\n", fahr, cels);
        fahr = fahr + step;
    }

    return 0;
}
```

Декларация (объявление) переменных.  
Они должны быть объявлены раньше,  
чем использованы в первый раз.

Операторы присвоения

Цикл. Пока верно **условие**,  
выполнять операторы  
между **фигурными**  
**скобками**

Печать по заданному формату

# Переменные

- Переменная - именованная область памяти, в которой хранится значение
- Имена переменных (идентификаторы)
  - ✓ a-z A-Z 0-9 \_ (не может начинаться с цифры)
  - X Пробельные символы
- Code style
  - ✓ massiveOfInteger
  - ✓ i\_outer
  - ✓ number\_of\_cows\_on\_the\_field
  - X jhg95
  - X \_KJY98jhg
  - X Vasya\_007

# Типы данных

- **целочисленные**
  - char (размер = 1 байту)
  - short
  - int
  - long
- **модификаторы**
  - signed
  - unsigned
- **дробные**
  - float
  - double

```
unsigned long int x;
```

# Арифметические операторы

- = присвоение
- == сравнение
- + сложение
- - вычитание
- \* умножение
- / деление
- % деление по модулю (взятие остатка)  
9%5 даст 4

# Краткая форма присвоения

- `fahr = fahr + step;`
- `fahr += step;`
- `fahr = fahr + 1;`  
`fahr += 1;`  
`fahr ++;`  
`++ fahr;`

# Форматирующие последовательности printf и scanf

- %d десятичный вывод
- %o восьмеричный (octet)
- %h шестнадцатеричный (hex)
- %f %g дробный
- %6d – печатать в 6 позиций
- %.2f – печатать 2 знака после запятой
- %6.2f – печатать 2 знака после запятой в поле на 6 позиций

# Цикл for

и комментарии

```
for (/*инициализация*/ ;  
    /*условие продолжения цикла*/ ;  
    /*итерация*/)  
{  
    /*тело цикла*/  
}
```

```
for (i = 0; i < N; i++) //среднестатистический цикл for  
{  
    a += b + c;  
    b *= i + c;  
}
```



# Цикл for

и комментарии

```
for (;;)           // можно ничего не писать, но  
                  // ; ставить обязательно
```

```
{  
    i++;  
    a += i;  
}
```

```
for (; i < 10; i++)  
    a *= i; //если оператор один, { } можно не ставить
```

```
for (i=10; i; i--); //так тоже можно
```

# Цикл for

пример

```
#include <stdio.h>
int main ( )
{
    int fahr, up = 300, low = 0, step = 20;

    for (fahr = low; fahr <= up ; fahr += step) {
        printf ("%3d %6.2f\n", fahr, 5 * (fahr - 32.0) / 9);
    }

    return 0;
}
```

# Именованные константы

- `#define LOW 0`  
`#define UP 300`  
`#define STEP 20`
- `#define HELLO_STR "Hello, world\n"`
- `fahr += STEP;`

# Ввод с клавиатуры

- `scanf` – стандартная функция ввода из библиотеки `stdio.h`
- Пример использования в коде:

```
int i, j;  
float x;  
scanf("%d", &i);  
scanf("%d%f",&j,&x);
```
- Не рекомендуется писать в формирующей строке что-либо кроме формирующих символов.
- Перед названием каждой передаваемой переменной необходимо писать амперсанд (&).

# Оператор if-else

- **if** (выражение)  
    инструкция\_1 ;  
**else**  
    инструкция\_2 ;

```
if ( a > b )  
    z = a;  
else  
    z = b;
```

- проверяется **выражение**
- если оно истинно (**не равно 0**), то выполняется инструкция\_1
- иначе выполняется инструкция\_2
- часть **else** можно опустить

# Операторы сравнения

- == равно
- != не равно
- <
- >
- <=
- >=
- пробелы внутри оператора не ставятся
- = это оператор "присвоить"!  
Не путайте с оператором сравнения!

# Логические операторы

- **&&** логическое И
- **||** логическое ИЛИ
- **!** отрицание

<b>&amp;&amp;</b>	false	true
false	false	false
true	false	<b>true</b>

<b>  </b>	false	true
false	false	<b>true</b>
true	<b>true</b>	<b>true</b>

```
if ( c == 'a' || c == 'A' )  
    na ++;
```

```
for (; c == 'a' || c == 'A'; )  
    na ++;
```

# Подсчет слов

```
#include <stdio.h>
int main () {
    char c, nl = 0;
    while ( (c = getchar() ) != '\n' ) {
        if (c == ' ')
            nl ++;
    }
    printf ("%d \n", nl);
    return 0;
}
```

char хранит целое число и состоит из одного байта. По таблице ASCII его можно расшифровать как символ. Форматирующая последовательность %c

Функция считывания нажатого на клавиатуре символа

Между кавычками стоит пробел



# Отладка программы

- Ошибки
  - синтаксические *syntax errors*
    - Забыли ; в конце выражения
  - выполнения *runtime errors*
    - Деление на 0
  - семантические *semantic errors*
    - (работает, но делает не то, что нужно)
- Отладочная печать
- Дебагер

# Задачи

- Вывести n-е число Фибоначчи. Ввод n с клавиатуры.
- С помощью печати \* на строке вывести
  - прямоугольник
  - треугольник
  - рамочку

Размеры фигуры вводить с клавиатуры.

```
putchar('*');    // печатает *
```

- Вывести на экран все множители числа n. Ввод n с клавиатуры.
- С экрана вводится последовательность нулей и единиц до введения любого другого символа. Подсчитать количество единиц.
- Вывести на экран ASCII-код цифры 0.
- Определить количество цифр во введенной последовательности символов.
- Вывести n-е простое число. Ввод n с клавиатуры.