

Страница курса <https://kappa.cs.petrsu.ru/~aksenova/>

Информатика и программирование

Информатика — наука о методах и процессах сбора, хранения, обработки, передачи, анализа и оценки информации, обеспечивающих возможность её использования для принятия решений.

"Computer Science" — наука о вычислениях.

Программирование — создание компьютерной программы для выполнения определенных вычислений.

Парадигма программирования — совокупность идей и понятий, определяющих стиль написания компьютерных программ.

Будем использовать язык программирования С/С++.

Первая программа :

```
/**  
 * main.c -- программа "Hello, students!"  
 *  
 * Copyright (c) 2022, Mikhail Kryshen <kryshen@cs.petrsu.ru>  
 *  
 * This code is licensed under MIT license.  
 */  
  
#include <stdio.h>  
int main()  
{  
    /* Выводим приветствие */  
    fprintf(stdout, "Hello, students!\n");  
    return 0;  
}
```

Этапы сборки программы

1) Препроцессор — подвергает код программы различным текстовым преобразованиям до трансляции в объектный.

Makefile и утилита make

```
# цель по умолчанию (при вызове make или make task1)
# собираем программу task1 из объектного файла task1.o
task1: main.o
        gcc -g -O0 -o task1 main.o
main.o: main.c
        gcc -g -O0 -c main.c
# цель clean (при вызове make clean)
# удаляем программу и объектные файлы
clean:
        rm task1 *.o
```

Структура программы

Тело функции

- операторы определения и описания переменных
- операторы-выражения

константы

переменные

операции

вызовы функций

- операторы потока управления (структурное программирование)

ветвления

циклы

блочный (составной) оператор

Переменные в программе С/С++

- модифицируемые объекты данных
- идентифицируются именем
- должны быть явно отнесены к некоторому типу

Имя переменной:

- начинается с латинской буквы или символа подчеркивания
- содержит латинские буквы, цифры или символ подчеркивания

Тип данных определяет

- формат представления значений объекта данных
- диапазон допустимых значений объекта данных
- диапазон допустимых операций

Определение переменной:

тип имя1, имя2, ...; тип имя = значение;

Диапазоны значений типов

Тип	Диапазон значений	Размер (байт)
bool	true и false	1
signed char	-128 ... 127	1
unsigned char	0 ... 255	1
signed short int	-32 768 ... 32 767	2
unsigned short int	0 ... 65 535	2
signed long int	-2 147 483 648 ... 2 147 483 647	4
unsigned long int	0 ... 4 294 967 295	4
float	3.4e-38 ... 3.4e+38	4
double	1.7e-308 ... 1.7e+308	8
long double	3.4e-4932 ... 3.4e+4932	10

Целочисленные типы

char

Формат scanf

%hhd, %c (как символ)

short

%hd

int

%d

long

%ld

С плавающей точкой

float

%f

double

%lf

```
#include <stdio.h>
#define PI 3.14159265358979323846

int main()
{
    /* Радиус круга */
    float r;

    /* Запрашиваем у пользователя радиус круга */
    fprintf(stdout, "Введите радиус: ");

    fscanf(stdin, "%f", &r);

    /* Вычисляем и печатаем площадь круга */
    fprintf(stdout, "Площадь круга: %.15f\n", PI * r * r);
    return 0;
}
```

```
#include <stdio.h>
#define PI 3.14159265358979323846

int main()
{
    /* Радиус круга */
    float r;

    /* Запрашиваем у пользователя радиус круга */
    printf("Введите радиус: ");

    scanf("%f", &r);

    /* Вычисляем и печатаем площадь круга */
    printf("Площаль круга: %.15f\n", PI * r * r);
    return 0;
}
```

Текстовый и двоичный поток

- Текстовый (**text**) поток - это последовательность символов
- Двоичный (**binary**) поток - это последовательность байтов

Способы использования файла

- **r** - открыть существующий файл для чтения (read)
- **w** - создать новый файл для записи (write)
- явно указать тип файла: **t** - текстовый; **b** - двоичный (бинарный)

```
FILE *fp; // файловая переменная
```

```
fp = fopen("my_letter.txt", "rt");
```

Вывод/ввод символа

- Функция **putc()** записывает символ в файл и имеет следующий прототип:

```
int putc(int c, FILE *fp);
```

- Функция **getc()** читает символ из файла и имеет следующий прототип:

```
int getc(FILE *fp);
```

Запись символа в файл

```
#include <stdio.h>
int main()
{
FILE *fp;    // файловый указатель
char c;      // Рабочая переменная
fp = fopen("my_text.txt", "wt"); // Файл открыли для
                                // записи
c = getchar();    // Ввод символа с клавиатуры
putc(c, fp);     // Запись символа в файл
fclose(fp);      // Закрытие файла
return 0;
}
```

Чтение символов из файла

```
FILE *fp;          // файловый указатель
char c;           // Рабочая переменная
fp = fopen("my_text.txt", "rt"); // Файл открыли для чтения
c = getc(fp);     // Прочитали первый символ из файла
while(c != EOF)   // Пока не прочитан символ конца файла
{
    putchar(c);   // Вывод символа на экран
    c = getc(fp); // Чтение следующего символа
}
fclose(fp);       // Закрытие файла
```

Создание файла символов

Символы вводятся с клавиатуры, точка – окончание ввода,
в файл не записывается

```
FILE *f;  
char z;  
if ((f = fopen("my_letter.txt", "wt")) == NULL)  
{    puts("Открыть файл не удалось\n");  
    return 0; }  
z = getchar();  
while( z != '.') // Пока символ не точка  
{  
    putc(z, f); // Запись символа в файл  
    z = getchar(); // Чтение след. символа  
}  
fclose(f); // Закрытие файла
```

Обработка символов во время чтения файла

Пусть файл содержит любые символы, в том числе звездочки.

Вывести на экран все символы, кроме звездочек.

```
FILE *f;  
char c;  
f = fopen("my_text.txt", "rt");  
c = getc(f);      // Прочитали первый символ из файла  
while(c != EOF)  // Пока не прочитан символ конца файла  
{  
    if( c != '*' ) putchar(c);      // Если не звездочка  
    c = getc(f);  
}  
fclose(f);
```

Работа с двумя файлами

Получение копии файла

```
FILE *i, *o;      // Понадобятся два файловых указателя
char c;           // Рабочая переменная
i = fopen("my_tex1.txt", "rt"); // Файл открыли для чтения
o = fopen("my_tex2.txt", "wt"); // Файл открыли для записи
c = getc(i);       // Прочитали первый символ
while(c != EOF)
{
    putc(c, o);   // Запись символа в новый файл
    c = getc(i);   // Чтение следующего символа
}
fclose(i);
fclose(o);
```

Сочетание операции чтения из файла с обнаружением конца файла

Первый способ:

```
c = getc(fp);  
while(c != EOF)  
{  
    // Обработка символа  
    c = getc(fp);      // Чтение следующего символа  
}
```

Второй способ:

```
while( (c = getc(fp)) != EOF )  
{  
    // Обработка символа  
}
```

Обработка символов во время чтения файла

```
FILE *f;  
char c;  
f = fopen("my_text.txt", "rt");  
    while( !feof() ) // Пока не достигнут конец файла  
{  
    c = getc(f);  
    if( c != '*' ) putchar(c);  
}  
fclose(f);
```