Простейший ввод-вывод в C

`stdio.h` — заголовочный файл (объявления) функций стандартной библиотеки ввода/выводв (язык C)

Концепция:

— Канал ввода/вывода. Должен быть открыт перед началом операций и закрыт после их окончания. Режимы открытия — чтение, запись, добавление.

Предвариательно открытые (системой по умолчанию) каналы — stdin (клавиатура), stdout (экран), stderr (экран). Их можно перенаправить в другие места (например, файлы).

Переменная для идентификации канала имеет специальный тип FILE*.

— Последовательнй доступ к каналу. Текущая позиция чтения/записи, она смещается после каждой операции на количество прочитанных/записанных символов (байтов).

— Форматный ввод/вывод. Библиотечные функции, как правило, имеют параметры, определяющие 1) канал 2) формат, т.е. способ формирования выводимых/вводимых символов из текстового представлния во внутренне представления числовых данных (или обратно) 3) список выводимых значений или указатели на место размещения вводимых значений.

— Есть режимы и функции бесформатного чтения/записи (т.е. без преобразований формы представления). О них разговор позже.

```
Функции stdio
    fopen      fclose
    fscanf     scanf     sscanf    и т.п.
    fprintf    printf    sprintf   и т.п.

Спецификации преобразования stdio
      %d  %hd %ld  (decimal)   int short long
      %u           (unsigned)
      %x  %X       (hexadecimal)
      %c           (char)
      %s           (string)
      %f  %lf      (float, double)
      %e  %le      (exponential)
      %g  %lg
      ...
      %N...        ширина
      %.M          точность
      %N.M                      %12.7f  %4d   %04d   %06x

     \n
     \t  \r
     \' \"  %%
```

Есть и другие — см. в справочниках

ВАЖНО! Форматная строка полностью определяет вид ввода и вывода. При этом все символы, которые указаны в формате ввода помимо спецификаций преобразования должны также появиться на вводе (помимо символов для значений вводимых переменных). Пробельные символы при вводе игнорируется (пропускаются).

Некоторые примеры использования функций ввода/вывода

```
#include <stdio.h>

void InOutInteger();
void InOutSignUnsign();
void InOutReal();
void OutWidthPrecision();
int  InOutFile();
void InOutString();

int main()
{
    InOutInteger();
    InOutSignUnsign();
    InOutReal();
    OutWidthPrecision();
    InOutFile();
    InOutString();

    return 0;
}

void InOutInteger()
{
    char a;
    short b;
    long c;

    printf("C: input char short long: ");
    scanf("%d%hd%ld", &a, &b, &c);
    printf("C: char %d short %d long %d\n", a, b, c);
}

void InOutSignUnsign()
{
    int d;
    unsigned int e;

    printf("C: input signed and unsigned: ");
    scanf("%d%u", &d, &e);
    printf("C: signed %d unsigned %u\n", d, e);
}

void InOutReal()
{
    float x;
    double y;

    printf("C: input float  double: ");
    scanf("%f%lf", &x, &y);
```

```
    printf("C: float %f double %f\n", x, y);
}

void OutWidthPrecision()
{
    float x = 3.141593;
    double y = 2.718281828459045;
    int z = 12345;

    printf("C: default %f %f\n", x, y);
    printf("C: precision  %.15f %.12f\n", x, y);
    printf("C: width.prec %16.5f %16.5f\n", x, y);
    printf("C: %.5e %20.16e\n", x, y);
}

int  InOutFile()
{
   int i, x, n=10;
   FILE *fin, *fout;

   // writing
   fout = fopen("input.txt","w");
   if (fout == 0) // или if (!fout)
   {
       printf("C: write opening error\n");
       return -1;
   }
   for (i=0; i<n; i++)
   {
       fprintf(fout,"%d\n", i);
   }
   fclose(fout);

   // reading
   fin = fopen("input.txt","r");
   if (fin == 0)  // или if (!fin)
   {
       printf("C: read opening error\n");
       return -1;
   }
   while (fscanf(fin,"%d", &x) == 1)
   {
       printf("%d ", x);
       //fprintf(stdout, "%d ", x);
   }
   fputc('\n', stdout);   // печать отдельного символа
   fclose(fin);

   return 0;
```

```
}


void InOutString()
{
    char str[256];

    printf("C: input word: ");
    scanf("%s",str);
    printf("C: you word: %s\n", str);
    printf("C: 3rd symbol: %c\n", str[2]);
}
```