

Понятие подпрограммы.
Процедуры и функции.

Во всех языках программирования существует возможность оформлять *часть программного кода* в форме *подпрограмм*.

Подпрограмма – это относительно самостоятельный фрагмент программы (блок), оформленный особым образом и имеющий имя.

Упоминание этого имени в программе называется **вызовом** подпрограммы.

Подпрограммы позволяют:

Во-первых, выполнять один и тот же программный код при различных значениях входящих в него **переменных - параметров,** открывают возможность повторного использования однажды созданного кода и способствуют появлению библиотеки подпрограмм.

Во-вторых, реализовать метод нисходящего проектирования программ (**метод «сверху-вниз»**), при котором программа разбивается на отдельные блоки-модули, каждый из которых выполняет некоторый логически завершённый этап обработки данных, а потом из этих блоков, как из «кирпичиков», собирается программа.

Различают подпрограммы-процедуры и подпрограммы-функции

(во многих языках программирования для этого используют зарезервированные слова **procedure** и **function**).

Функция отличается от *процедуры* двумя особенностями:

- всегда вычисляет некоторое значение, возвращаемое в качестве результата *функции*;
- вызывается как операнд в выражениях.

Процедура имеет свои особенности:

- может просто выполнять некоторую последовательность действий и не возвращать никакого результата (*отсутствие результата*);
- вызов *процедуры* является оператором языка;
- имеет *входные и выходные аргументы*, причем *выходных аргументов* – ее результатов - может быть достаточно много.

Хорошо известно, что одновременное существование в языке *процедур* и *функций* в каком-то смысле избыточно.

Добавив еще один *выходной аргумент*, любую *функцию* можно записать в виде *процедуры*. Справедливо и обратное. Если допускать *функции с побочным эффектом*, то любую *процедуру* можно записать в виде *функции*.

Однако значительно удобнее иметь обе формы реализации подпрограмм: и *процедуры*, и *функции*. Обычно подпрограмму предпочитают реализовать в виде *функции* тогда, когда она имеет один *выходной аргумент*, рассматриваемый как результат вычисления значения *функции*.

Возможность вызова *функций* в выражениях также влияет на выбор в пользу реализации подпрограммы в виде *функции*. В других случаях подпрограммы реализуют в виде *процедуры*.

Для ОО-языков, к которым относится и язык С#, программная система строится из модулей, роль которых играют *классы*. **Процедуры и функции** связываются теперь с **классом** и называются **методами класса**.

То есть в С# **процедуры и функции** существуют только как **методы некоторого класса**, они не существуют вне **класса**.

Прежнюю роль библиотек *процедур и функций* теперь играют *библиотеки классов*. Библиотека классов FCL, доступная в языке С#, существенно расширяет возможности языка.