|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Когда программа запускается – вызывается (запускается) функции ***main()*** | | | | | | | | | | | | | | | | | | | |  |  |  |
| Функции и конструкции ***if***-***else***, ***for***, ***while*** и ***do***-***while*** имеют свое тело(блок) ***{ }*** | | | | | | | | | | | | | | | | | | | | | |  |
| То, что создано внутри блока, за его пределами перестает существовать | | | | | | | | | | | | | | | | | | | |  |  |  |
| Проверка: | | | | | | |  | Циклы: | | | | | | | | | | | | | |  |
| *if ( )*  *{ }*  *else*  *{ }* | | *если (условие)*  *{ то }*  *иначе*  *{ это }* | | | | |  | *while ()*  *{ }* | | | *пока (условие)*  *{ повторять это }* | | | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | *do { }*  *while ()****;*** | | | *делать { это }*  *пока (условие)****;*** | | | | |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Подключение файлов: | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | библиотечных: | | | | | |  | *for (*int i = 0***;***i < max***;***i++*)*  *{ }* | | | | | | | max раз  { повторять это } | | | | | |  |  |
|  | *#include <* имя *>* | | | | | |  |  |  |
|  | своих: | | | | | |  |  |  | для i от 0 до max, увеличивая i на 1  { повторять это } | | | | | | | | | |  |  |  |
|  | *#include “* имя *“* | | | | | |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Точка с запятой ставится в конце любого выражения — в конце  почти каждой строчки кода, кроме ***{ }*** и расписанных выше конструкций | | | | | | | | | | | | | | | | | | | |  |  |  |
|  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| Тип переменной указывается только при ее создании: | | | | | | | | | | | | | | | | | |  |  |  |  |  |
|  | *int* a***;*** | | | — | целое число (-2, -1, 0, 1 , 2) | | | | | | | | | | | | |  |  |  |  |  |
|  | *bool* b***;*** | | | — | логическое значение (true, false) | | | | | | | | | | | | |  |  |  |  |  |
|  | *char* c***;*** | | | — | символ (‘f’, ‘F’, ‘v’, ‘ы’, ‘3’, ‘/n’) | | | | | | | | | | | | |  |  |  |  |  |
|  | *double* d***;*** | | | — | число с плавающей точкой (3.14, 2.83, 0.44) | | | | | | | | | | | | |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | При создании массива указывается его размер: | | | | | | | | | | | | | |  |  |  |  |  |  |  |  |
|  | *int* mas*[*100*]****;*** | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | *char* str*[*20*]****;*** | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Обратиться к элементу массива можно через его номер (индекс): | | | | | | | | | | | | | | | | | | |  |  |  |
|  | mas*[*0*]* = 1***;*** | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | str*[*19*]* = ***‘***К***’;*** | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  | Константные значения в коде: | | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |
|  |  | Числа – просто: | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |
|  |  | *int* k = 10***;*** | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | Символы – в одинарных кавычках: | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |
|  |  | *char* k2 = ***‘***S***’;*** | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  | Строки – в двойных кавычках: | | | | | | | | | |  |  |  |  |  |  |  |  |  |  |  |
|  |  | *char* k3*[*10*]* = *“*string*”****;*** | | | | | | |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |