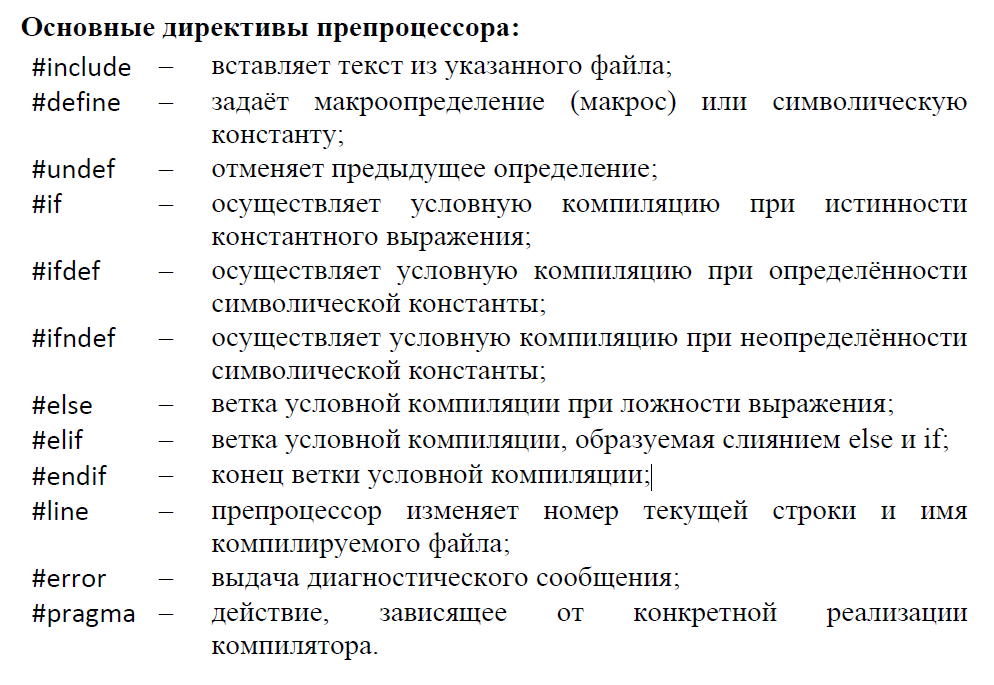
|  |  |
| --- | --- |
| 1) ***Препроцессор*** – программа для обработки текста.   |  | | --- | | Входные и выходные данные для препроцессора имеют ***текстовый формат***.  Препроцессор преобразует текст в соответствии с ***директивами препроцессора***.  2) директивы: #include, #define, #if, #else, #elif, #endif, #ifdef, #ifndef, #error, #line, #pragma, #undef; операторы: defined, #, ##.  3)Оперативная память, стек, куча, кеш…(?)  4)Пространство имен — это декларативная область, в рамках которой определяются  различные идентификаторы (имена типов, функций, переменных,  и т. д.).  5) В языке C++ **исключение** – **это** специальный объект класса или значение базового типа, который описывает (определяет) конкретную исключительную ситуацию и соответствующим образом обрабатывается. (try-catch) |   6) throw (в переводе — обработать, запустить), try (попытка), catch (поймать, ловить).В исходном коде ниже, исключение сработает так: программа получает конкретное указание от программиста — если значение определённой переменной в определённом участке кода (в try-блоке) будет равно 0(например), то в этом случае пусть генерируется исключение throw. Это исключение автоматически передастся catch-блоку в виде параметра и выполнится код этого блока. |

7) **Необработанное** **исключение** остается для обработки вызывающей **функцией**. Это продолжается до тех пор, пока **исключение** не будет обработано. Это происходит с **вызовом** **функции** или без него. Другими словами, если вызывается **функция**, которая не находится в блоке try, **исключение**, которое происходит в этой **функции**, автоматически передается в **стек** **вызовов**.

8)Статические переменные(static…)

9)Выделение динамической памяти(динамические массивы, структуры…)

10)Все переменные, которые хранятся в main-е

11)

12)#if, #ifdef, #ifndef, #else, #elif, #endif(конец ветки усл компиляции)

13)namespace <name> // using namespace <name> (по умолчанию)

14) == 6