|  |  |
| --- | --- |
| **Определение** | **Что это?** |
| int\* ptr; | Указатель на int.  Значение указателя – адрес объекта типа int.  Разыменовав такой указатель получим доступ к объекту типа int. |
| int\*\* ptr; | Указатель на int\*  Значение указателя – адрес объекта типа int\*  Разыменовав такой указатель получим доступ к объекту типа int\* |
| int\*\*\* ptr; | Указатель на int\*\*.  Значение указателя – адрес объекта типа int\*\*.  Разыменовав такой указатель получим доступ к объекту типа int\*\* |
| const int \* ptr; | Указатель на константу типа int.  Значение указателя – адрес объекта типа const int.  Разыменовав такой указатель получим доступ к объекту типа const int. Объект после разыменования ***нельзя*** будет изменить.  Само значение указателя ***можно*** изменить. |
| int \* const ptr = NULL; | Константный указатель на int.  Значение указателя – адрес объекта типа int.  Разыменовав такой указатель получим доступ к объекту типа int. Объект после разыменования ***можно*** будет изменить.  Само значение указателя ***нелья*** изменить, поскольку сам указатель – это константа.  Требуется инициализация при определении. |
| const int \* const ptr = NULL; | Константный указатель на константу типа int.  Значение указателя – адрес объекта типа const int.  Разыменовав такой указатель получим доступ к объекту типа const int. Объект после разыменования ***нельзя*** будет изменить.  Само значение указателя ***нелья*** изменить, поскольку сам указатель – это константа.  Требуется инициализация при определении. |
| int\* ptr[10]; | Массив указателей на int.  Массив из 10 элементов. Каждый элемент – переменная типа int\*.  Выражение ptr + 2 в результате даст новый адрес, который будет больше исходного ptr на 2 \* sizeof(int\*) |
| int (\*ptr) [10]; | Указатель на массив из 10 элементов типа int.  Выражение ptr + 2 в результате даст новый адрес, который будет больше исходного ptr на 2 \* sizeof(int[10])  (т.е новый адрес больше на 80)! |
| const int (\* const ptr) [10] = NULL; | Константный указатель на массив из 10 элементов типа const int.  Разыменовав такой указатель получим доступ к массиву элементов const int. Массив после разыменования ***нельзя*** будет изменить.  Само значение указателя ***нелья*** изменить, поскольку сам указатель – это константа.  Требуется инициализация при определении.  Выражение ptr + 2 в результате даст новый адрес, который будет больше исходного ptr на 2 \* sizeof(int[10])  (т.е новый адрес больше на 80)! |
| **Определение** | **Что это?** |
| double \* (\*ptr) [10]; | Указатель на массив из 10 элементов типа  double \*.  Выражение ptr + 2 в результате даст новый адрес, который будет больше исходного ptr на 2 \* sizeof(double \*[10])  (т.е новый адрес больше на 80)! |
| double (\*ptr) (int); | Указатель на функцию, которая возвращает double и принимает 1 аргумент типа int |
| double \*ptr (int); // Объявление | Объявление (прототип) функции, которая возвращает double \* (указатель на double) и принимает 1 аргумент типа int |
| void (\*ptr) (void); | Указатель на функцию, которая возвращает void (т.е. ничего) и не принимает аргументов |
| double (\*ptr[10]) (int); | Массив из 10 элементов. Каждый элемент массива – это:  указатель на функцию, которая возвращает double и принимает 1 аргумент типа int |
| double \* (\* (\*ptr) [10]) (int); | **Указатель на массив** из 10 элементов.  Каждый элемент массива, на который указывает указатель это:  *указатель на функцию, которая*  *возвращает double \* и принимает 1 аргумент типа int.*  **Указатель на массив указателей на функцию.**  Выражение ptr + 2 в результате даст новый адрес, который будет больше исходного ptr на 2 \* sizeof(double \*(\*[10]) (int))  (т.е новый адрес больше на 80)! |

**Примечание**: все уточнения касательно числовых значений адресов корректны только для 32-разрядного режима.