Ответы на вопросы

1. Шесть модификаторов доступа и для чего они используются:

|  |  |
| --- | --- |
| Модификатор доступа | Описание |
| private | Доступ ограничен содержащим типом. Используется по умолчанию |
| internal | Доступ ограничен содержащим типом и любым другим типом в текущей сборке |
| protected | Доступ ограничен содержащим типом или типами, производными от содержащего типа |
| public | Неограниченный доступ |
| internal protected | Доступ ограничен содержащим типом и любым другим типом в текущей сборке, а также типами, производными от содержащего класса. Аналогичен вымышленному модификатору доступа internal\_or\_protected (то есть дает доступ по принципу internal ИЛИ protected) |
| Private protected | Доступ ограничен содержащим типом и любым другим типом, который наследуется от типа и находится в той же сборке. Аналогичен вымышленному модификатору доступа internal\_and\_protected (то есть дает доступ по принципу internal И protected). Эта комбинация доступна только для версии C# 7.2 или более поздних |

Статический член – определённое поле, в котором есть только одно значение, общее для всех экземпляров, при этом поля не единственные члены, которые могут быть статическими. Для обозначения статического члена класса используется модификатор static.

Константы - поля, которые никогда не должно меняться. Компилятор буквально копирует данные в любой код, который их читает. Для обозначения константы класса используется модификатор const.

Статические поля readonly (только для чтения), значения которых будут общими для всех экземпляров типа.

1. Конструкторы — это особая категория методов.

Конструкторы выполняются, когда используется ключевое слово new для выделения памяти и создания экземпляра класса

1. [Flags] применяется для комбинирования вариантов в одном знакчении.
2. Кл. слова partial разделяет определения сложных классов на несколько файлов.
3. –
4. Кл. слово ref используется не только для передачи параметров в метод, но и для возвращаемого значения. Это позволяет внешней переменной ссылаться на внутреннюю и изменять ее значение после вызова метода. Это может быть полезно в сложных сценариях, например для передачи пустых контейнеров, требующих последующего заполнения, в больших структурах данных, и выходит за рамки книги
5. Перезагрузка методов: Вместо двух методов с разными именами мы могли бы присвоить обоим методам одно и то же имя. Это допустимо, поскольку каждому методу присуща собственная сигнатура.

Поля, доступные только для чтения, — данные в таких полях не могут измениться после создания экземпляра класса, но могут быть рассчитаны или загружены из внешнего источника во время создания экземпляра

Свойства выполняются, когда необходимо получить доступ к данным. Они обычно хранятся в поле, но могут храниться извне или рассчитываться во время выполнения

1. Определить параметр как необязательный, указывая значение по умолчанию в списке параметров метода. Необязательные параметры всегда должны быть последними в списке параметров.