Модуль 8: Создание и разрушение объектов

Объекты и память

- Объекты и область их видимости
- Garbage Collection (сборка мусора)

Объекты и область их видимости

- **Время жизни локальной переменной, определяется областью видимости, в которой она определена**
  - Недолгое время существования (как правило)
  - Детерминистическое создание и уничтожения
- **Время жизни динамического объекта не привязано к области его видимости**
  - Более длительное существование
  - Недетерминированное уничтожение

Garbage Collection (сборка мусора)

- **Вы не можете явно уничтожать объекты**
  - В C# нет оператора, обратного **new** (например, **delete**)
  - В других языках программирования явный вызов оператора **delete** часто осуществлялся неверным образом
- **Сборщик мусора уничтожает объекты за вас**
  - Находит объекты, на которые не существует ни одной ссылки
  - Возвращает память для повторного использования
  - Активизируется только по необходимости

Управление ресурсами

- Освобождение ресурсов
- Деструкторы
- Момент вызова деструктора
- Интерфейс IDisposable и метод Dispose
- Инструкция using в C#

Освобождение ресурсов

- **Различные объекты завершают свою работу по-разному**
  - У объектов .NET Framework есть метод **Finalize**
  - Деструктор вызывается перед непосредственным уничтожением объекта
  - Для корректного освобождения ресурсов в C# используются деструкторы. Вы не можете вызвать или переопределить метод **Object.Finalize**.

Деструкторы

- **Деструктор – механизм для очистки ресурсов**
  - Особенности синтаксиса:
    - Нет модификатора доступа
    - Нет типа возвращаемого значения (даже не **void**)
    - Имя совпадает с именем класса, перед именем деструктора указывается знак ~
    - Нет параметров

```
class SourceFile
{
    ~SourceFile( ) { ... }
}
```

Момент вызова деструктора

- **Момент и порядок разрушения объектов не определены**
- **Деструктор обязательно будет запущен**
  - Время запуска не определено
- **По возможности избегайте использования деструкторов**
  - Увеличение нагрузки
  - Сложность
  - Откладывают освобождение памяти

Интерфейс IDisposable и метод Dispose

- **Для освобождения ресурсов:**
  - Реализуйте метод **Dispose** интерфейса **IDisposable,** предназначенный для освобождения ресурсов
  - Вызовете метод **GC.SuppressFinalize**
  - Убедитесь в том, что что в программе не используются освобожденные ресурсы
- **При реализации этого метода необходимо убедиться, что все занятые ресурсы высвобождены путем передачи вызова по иерархии вложений**

Лабораторная работа 8.2: Управление ресурсами