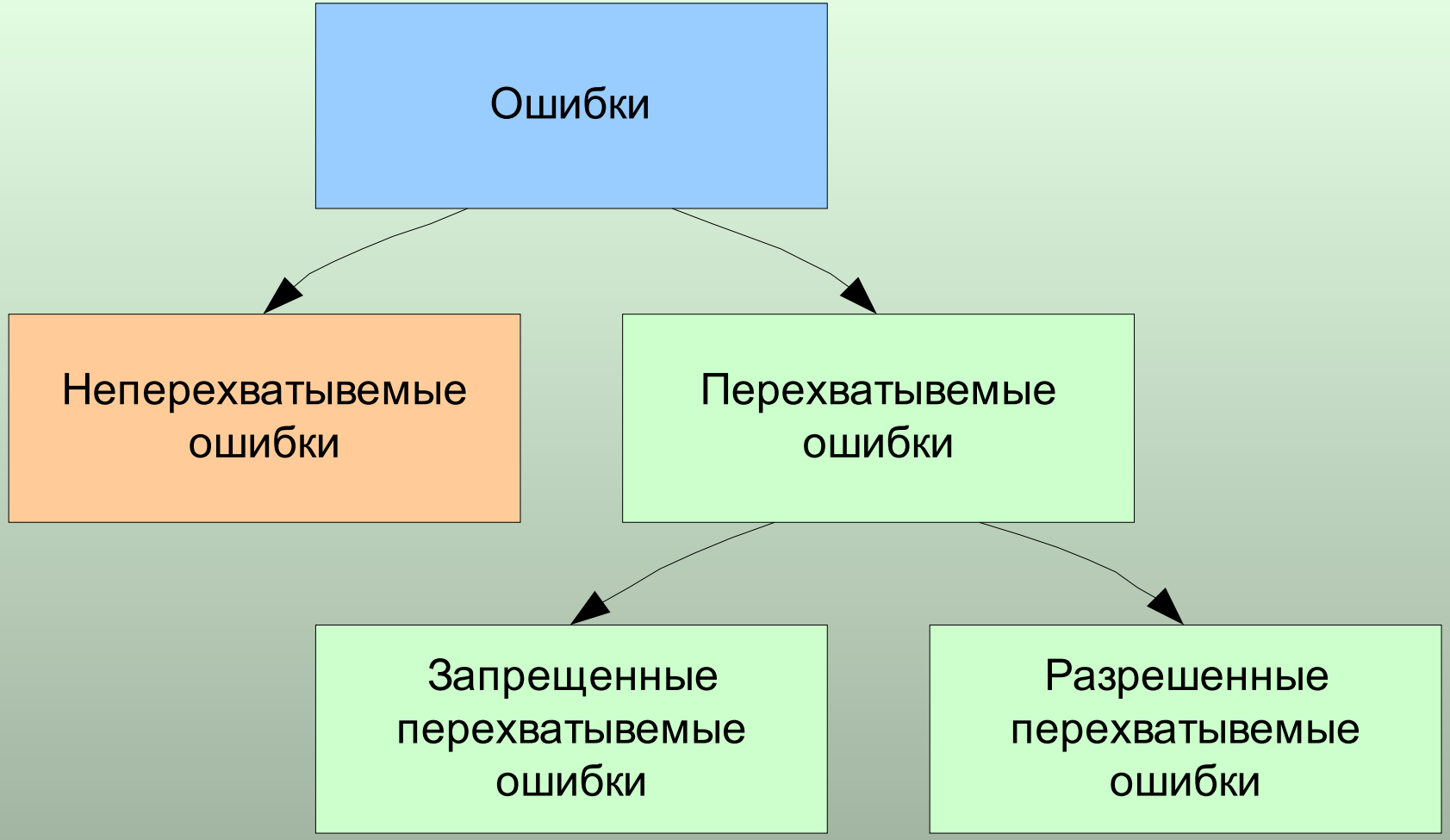
А.В. Макаров, С.Ю. Скоробогатов, А.М. Чеповский

Учебный курс “CIL и системное программирование в Microsoft .NET”

Лекция 2.

Общая система типов

Классификация ошибок в программах



Безопасный код и код, имеющий хорошее поведение

- Фрагмент программы, в котором не могут возникнуть неперехватываемые ошибки, называется безопасным (safe)
- Для любого языка программирования можно определить класс ошибок, называемых запрещенными (forbidden errors). В этот класс следует включить все неперехватываемые ошибки, а также некоторое подмножество перехватываемых ошибок
- Говорят, что фрагмент программы имеет хорошее поведение (well behaved), если в нем не могут возникать запрещенные ошибки

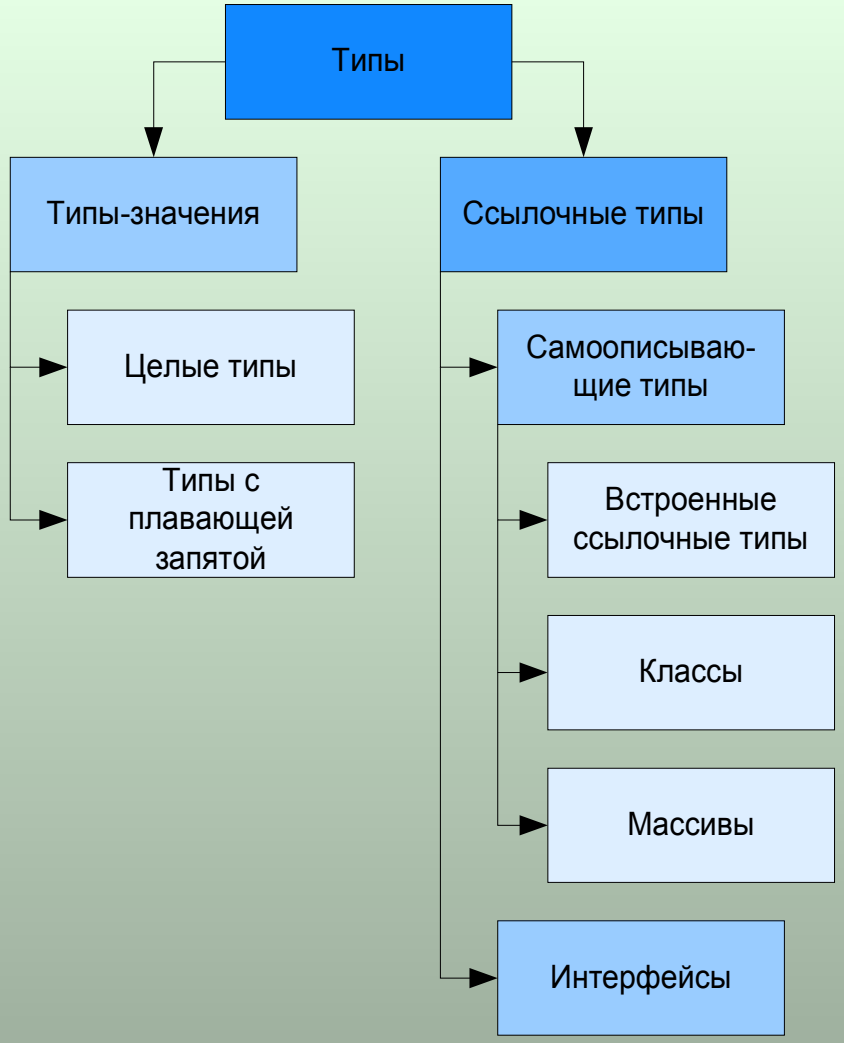
Предназначение системы типов

- Языки программирования, которые гарантируют хорошее поведение всех написанных на них программ, называются языками со строгой проверкой (strongly checked)
- Существуют два пути для диагностики запрещенных ошибок: статическая проверка программы до ее выполнения (static checking) и динамическая проверка во время выполнения (dynamic checking)
- Система типов в языке программирования разрабатывается для того, чтобы можно было осуществлять статическую проверку программы. Она представляет собой набор правил, определяющих условия, при которых конструкции языка не вызывают запрещенных ошибок

Общая система типов .NET

- Общая система типов (Common Type System – CTS) является объединением систем типов основных распространенных в настоящее время языков программирования
- Благодаря CTS, система выполнения может проводить верификацию двоичных исполняемых файлов непосредственно перед их запуском. Это гарантирует безопасность кода, выполняемого в среде .NET и, тем самым, обеспечивает возможность автоматического управления памятью (сборки мусора)

2.1. Ядро системы типов .NET



- Типы-значения представляют примитивные типы данных (целые числа и числа с плавающей запятой). Существуют еще пользовательские типы-значения, но мы обсудим их позже
- Ссылочные типы описывают так называемые объектные ссылки (object references), которые представляют собой адреса объектов

Ячейки

- Значения любого типа хранятся в ячейках (location). Для каждой ячейки известен тип значений, которые она может содержать
- В качестве ячеек могут выступать:
  - Глобальные переменные
  - Локальные переменные
  - Параметры методов
  - Поля объектов
  - Элементы массивов
- Ячейки не могут содержать объекты. Все объекты размещаются в специальной области памяти, называемой кучей (heap)

2.1.1. Встроенные типы-значения

Тип	Имя в .NET Framework Class Library	Описание
`bool`	`System.Boolean`	булевский (8 бит)
`char`	`System.Char`	символ Unicode (16 бит)
`int8`	`System.SByte`	целое со знаком (8 бит)
`int16`	`System.Int16`	целое со знаком (16 бит)
`int32`	`System.Int32`	целое со знаком (32 бит)
`int64`	`System.Int64`	целое со знаком (64 бит)
`unsigned int8`	`System.Byte`	целое без знака (8 бит)
`unsigned int16`	`System.UInt16`	целое без знака (16 бит)
`unsigned int32`	`System.UInt32`	целое без знака (32 бит)
`unsigned int64`	`System.UInt64`	целое без знака (64 бит)
`float32`	`System.Single`	вещественное (32 бит)
`float64`	`System.Double`	вещественное (64 бит)
`native int`	`System.IntPtr`	целое со знаком (разрядность процессора)
`native unsigned int`	`System.UIntPtr`	целое без знака (разрядность процессора)

2.1.2. Самоописывающие ссылочные типы

- В среде .NET объекты и объектные ссылки хранятся раздельно, а именно: объекты хранятся в куче, а ссылки – в ячейках. Один и тот же ссылочный тип может являться как типом объекта, так и типом объектной ссылки
- Каждый объект в куче содержит информацию о своем типе. Поэтому ссылочные типы, представляющие объекты, называются самоописывающими (self-describing)

Встроенные классы

- System.Object
  - Является общим базовым классом, от которого непосредственно или транзитивно наследует любой другой класс
- System.String
  - Используется для представления строковых данных в формате Unicode

Классы

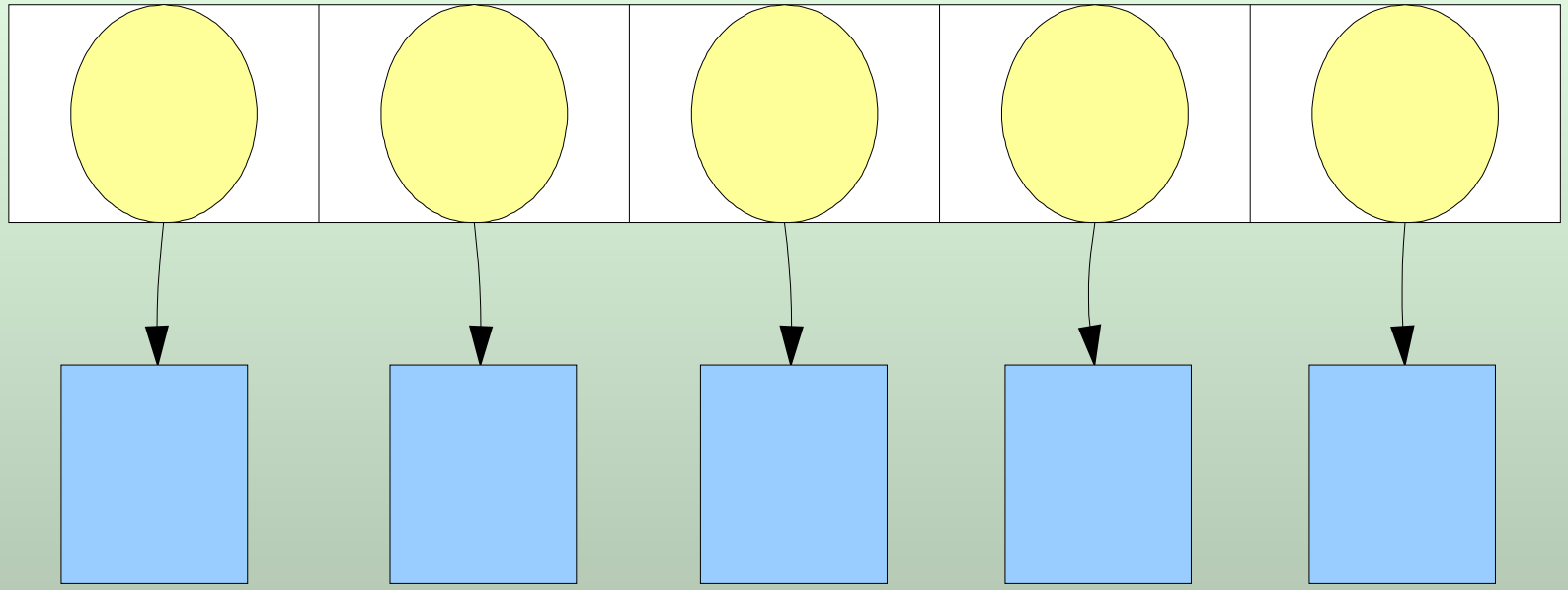
- Классы могут агрегировать значения других типов, а также наследоваться друг от друга (в .NET поддерживается только одиночное наследование)
- Элементы, которые могут содержаться в классах:
  - Поля (fields)
  - Методы (methods)
  - Свойства (properties)
  - События (events)

Типы-массивы

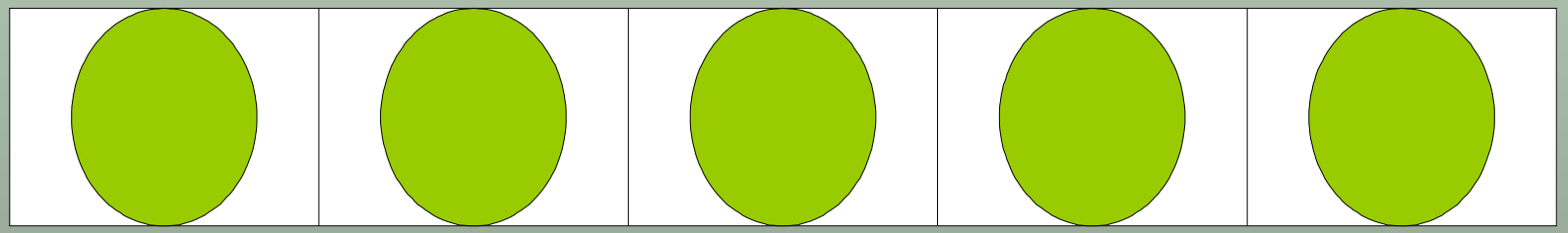
- Каждый массив представляет собой объект в куче, доступ к которому осуществляется через объектную ссылку
- Типы-массивы наследуют от System.Array
- Типы-массивы интересны тем, что в отличие от классов, определяемых программистом самостоятельно, формируются системой автоматически
- Массивы бывают как одномерными, так и многомерными. Кроме того, система поддерживает массивы, нижняя граница которых отлична от нуля

Особенности представления массивов

Массив ссылочного типа



Массив типа-значения



2.1.3. Типы-интерфейсы

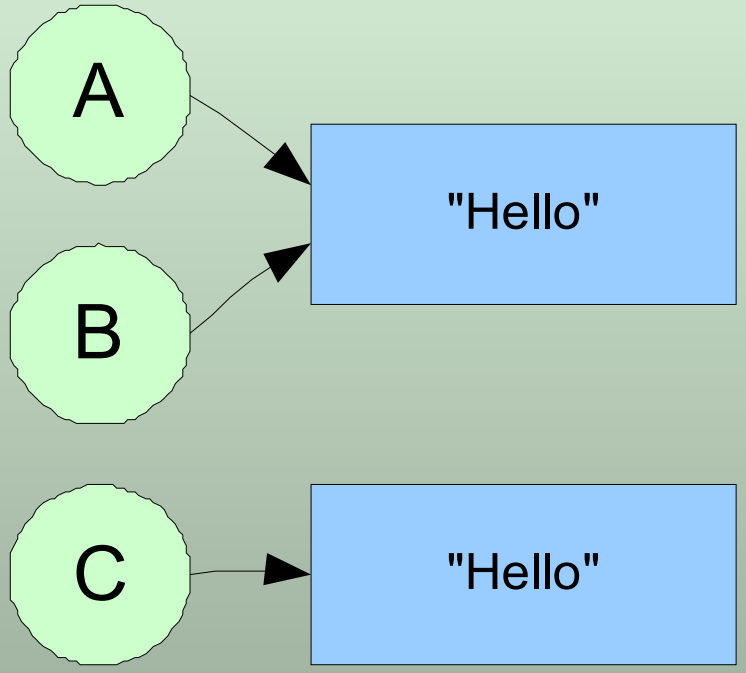
- Интерфейсы служат для компенсации отсутствия в .NET множественного наследования. Они могут рассматриваться как чисто абстрактные классы, содержащие только перечисленные ниже элементы:
  - Абстрактные методы
  - Статические методы
  - Статические поля
  - Абстрактные свойства
  - Абстрактные события

2.1.4. Совместимость ячеек по присваиванию

- Значение может иметь сразу несколько типов
- Ячейка может содержать только значение, совместимое с ее типом. То есть общая система типов не допускает присваивание ячейке несовместимого с ее типом значения
- Условие совместимости значения и ячейки: для того, чтобы некоторое значение можно было присвоить заданной ячейке, необходимо и достаточно, чтобы хотя бы один из типов значения совпадал с типом ячейки

2.1.5. Идентичность и равенство значений

- Для объектных ссылок и значений типов-значений вводятся отношения идентичности (identity) и равенства (equality)



A идентична B,
A не идентична C,
B не идентична C

A = B = C

2.2. Дополнительные элементы системы типов .NET

2.2.1. Структуры и перечисления

- Одной из основных причин ухудшения производительности является медленная работа сборщика мусора, вызванная большим количеством мелких объектов в куче. Это явление можно наблюдать в двух случаях:
  - Интенсивное создание временных объектов с очень малым временем жизни. Зачастую такие объекты создаются и используются в теле одного метода
  - Использование гигантских массивов объектов, при котором в этих массивах хранятся ссылки на огромное количество небольших объектов
- В общую систему типов были добавлены пользовательские типы-значения: структуры и перечисления

Структуры

- Структуры являются аналогом классов. Они, как и классы, могут содержать поля, методы, свойства и события
- Все структуры неявно наследуют от библиотечного класса System.ValueType, и, более того, встроенные типы-значения также наследуют от этого класса
- Структуры не могут наследоваться друг от друга, и, тем более, не могут наследоваться от классов (кроме System.ValueType)

Перечисления

- Перечисления представляют собой структуры с одним целочисленным полем Value
- Перечисления содержат набор констант, определяющих возможные значения поля Value. При этом для каждой константы в перечислении хранится ее имя
- Перечисления неявно наследуют от библиотечного класса System.Enum, который, в свою очередь, является наследником класса System.ValueType

2.2.2. Указатели

- Использование указателей может значительно увеличить производительность. Однако считается, что применение указателей чревато появлением в программах большого количества трудноуловимых неперехватываемых ошибок
- Разработчикам .NET удалось добавить указатели в общую систему типов. При этом появилось две категории указателей: управляемые указатели (managed pointers) и неуправляемые указатели (unmanaged pointers)
- Программа, в которой используются неуправляемые указатели, автоматически считается небезопасной и не может пройти верификацию

Управляемые указатели

- Для того чтобы программы оставались безопасными, на использование управляемых указателей наложен целый ряд ограничений:
  - Управляемые указатели могут содержать только адреса ячеек, то есть они могут указывать исключительно только на глобальные и локальные переменные, параметры методов, поля объектов и ячейки массивов
  - За каждым указателем закреплен тип ячейки, на которую он может указывать
  - Указатели могут храниться только в локальных переменных и параметрах методов
  - Запрещены указатели на указатели

2.2.3. Упакованные типы-значения

- Упакованные типы-значения являются ссылочными и самоописывающими. Объекты этих типов предназначены для хранения значений типов-значений
- Упаковка заключается в том, что в куче создается пустой объект нужного размера, а затем значение копируется внутрь этого объекта
- Распаковка заключается в том, что мы получаем управляемый указатель на содержимое объекта упакованного типа-значения