Creating Nested Classes

Вложенный класс - это класс, определенный в другом классе. Вложенный класс может иметь одну из четырех разновидностей.

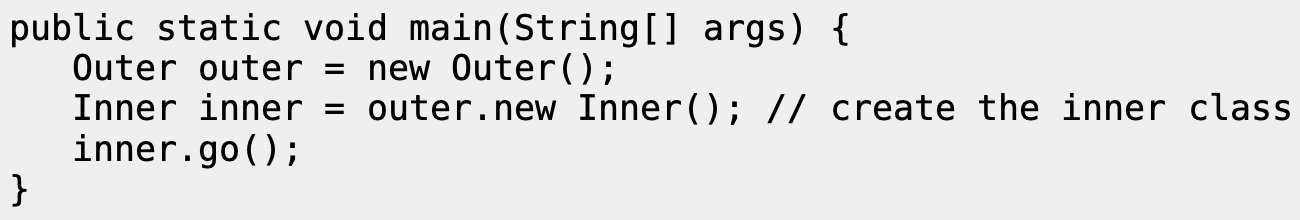
* Внутренний класс: нестатический тип, определенный на уровне члена класса.
* Статический вложенный класс: статический тип, определенный на уровне члена класса.
* Локальный класс: класс, определенный в теле метода.
* Анонимный класс: особый случай локального класса, у которого нет имени.

Declaring an Inner Class

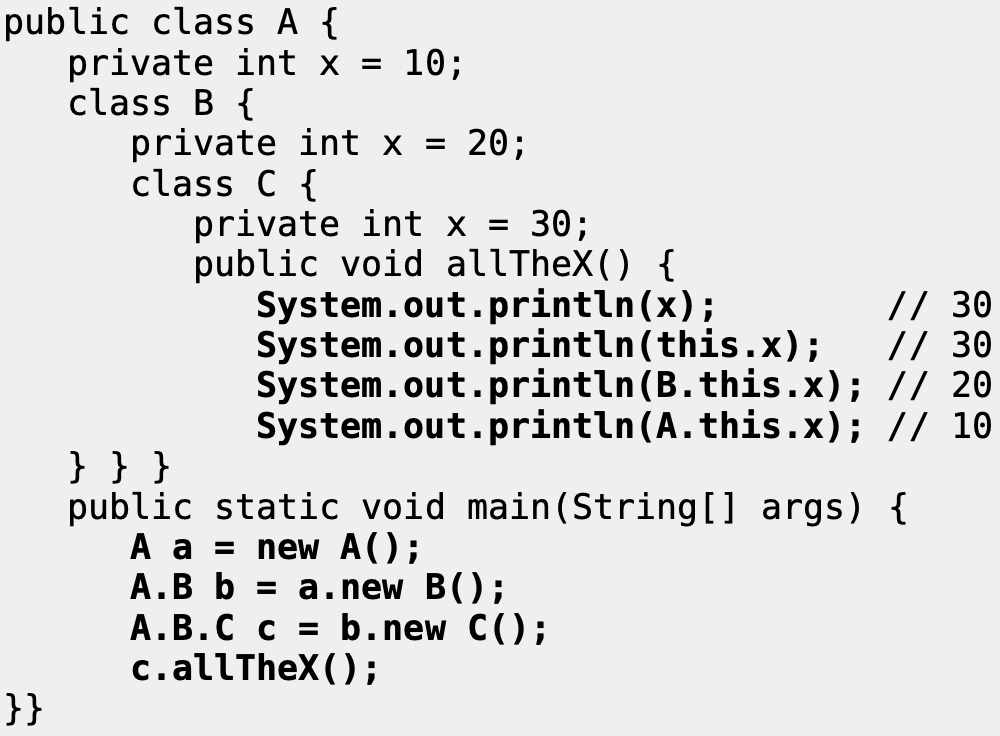
Inner класс, также называемый внутренним классом-членом, представляет собой нестатический тип, определенный на уровне члена класса (на том же уровне, что и методы, переменные экземпляра и конструкторы). Внутренние классы обладают следующими свойствами:

* Может быть объявлен public, protected, package-private (default), или private.
* Может расширять любой класс и реализовывать интерфейсы.
* Может быть отмечен как abstract или final
* Невозможно объявлять статические поля или методы, за исключением static final полей.
* Может получить доступ к членам внешнего класса, включая частные члены.

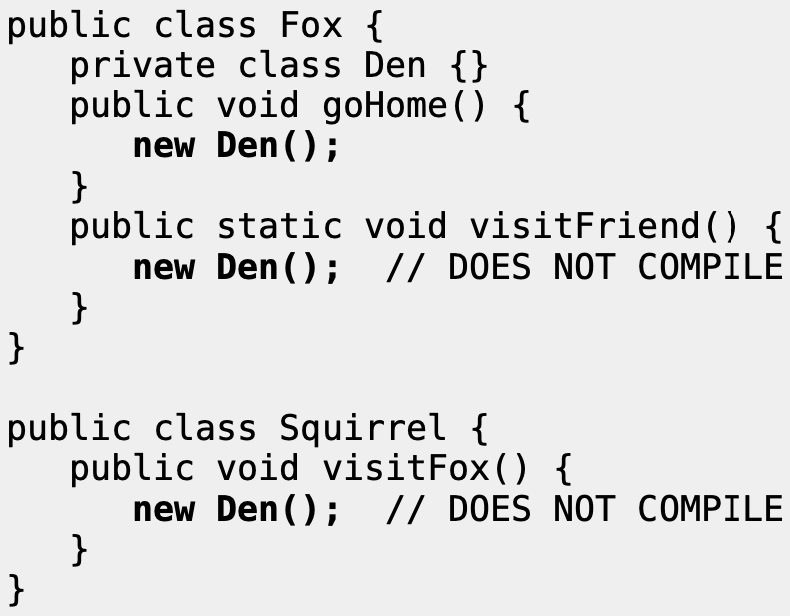
Создание внутреннего класса:



Доступ к переменным через this:



Внутренние классы требуют экземпляра. Внутренний класс не может быть создан без экземпляра в статическом контексте



Creating a static Nested Class

Static nested класс - это статический тип, определенный на уровне члена. В отличие от inner класса, static nested класс может быть создан без экземпляра включающего класса. Однако компромисс заключается в том, что он не может напрямую обращаться к переменным экземпляра или методам во внешнем классе. Это можно сделать, но для этого требуется явная ссылка на переменную внешнего класса.

Другими словами, это как класс верхнего уровня, за исключением следующего:

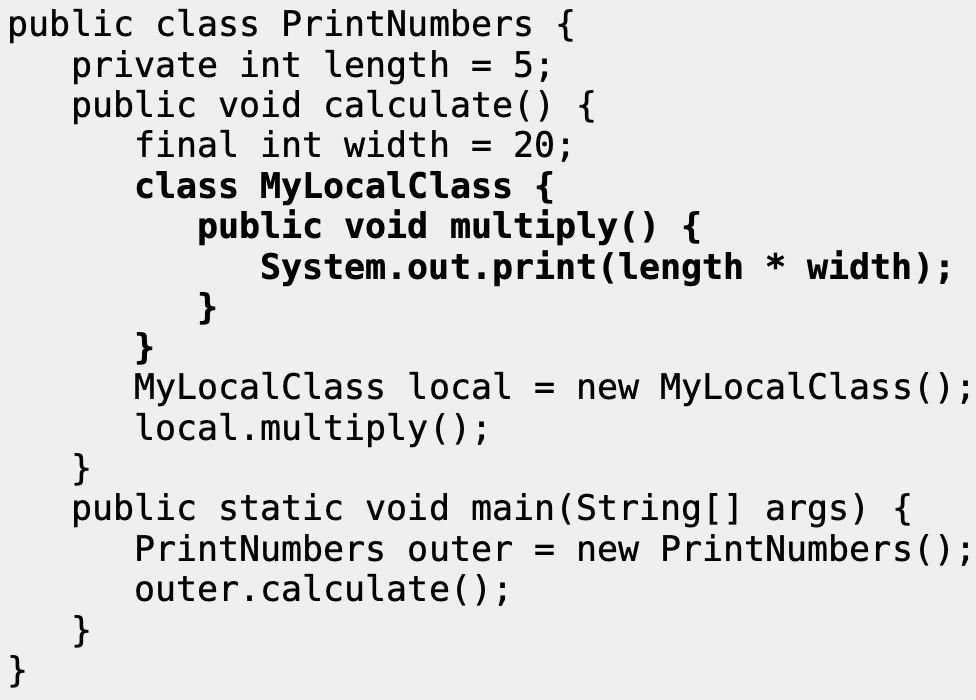
* Вложение создает пространство имен, потому что имя включающего класса должно использоваться для ссылки на него.
* Его можно сделать private или использовать один из других модификаторов доступа для его инкапсуляции.
* Вмещающий класс может ссылаться на поля и методы статического вложенного класса.
* Может содержать статические переменные.

Writing a Local Class

Локальный класс - это вложенный класс, определенный внутри метода. Как и локальные переменные, объявление локального класса не существует до вызова метода и выходит за пределы области видимости, когда метод возвращается. Это означает, что вы можете создавать экземпляры только внутри метода. Эти экземпляры все еще можно вернуть из метода. Так работают локальные переменные.

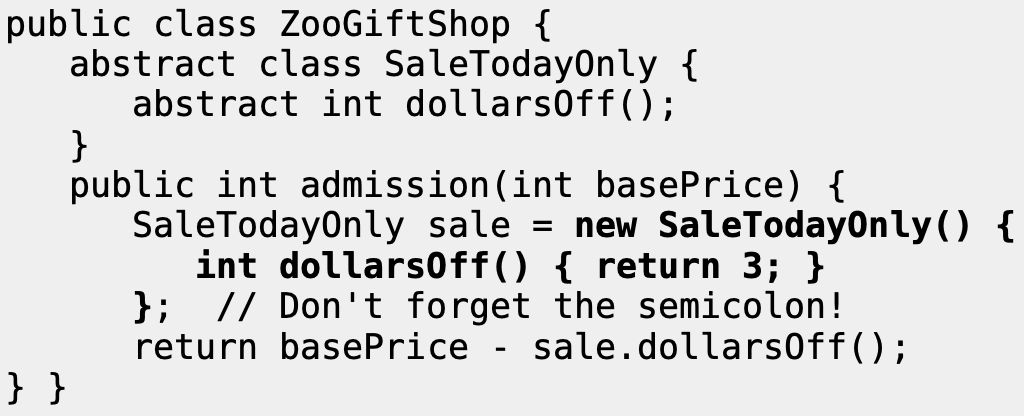
Локальные классы обладают следующими свойствами:

* У них нет модификатора доступа.
* Они не могут быть объявлены статическими и не могут объявлять статические поля или методы, за исключением статических конечных полей.
* У них есть доступ ко всем полям и методам включающего класса (если они определены в методе экземпляра).
* Они могут получить доступ к локальным переменным, если переменные являются final или effectively final. Ограничение из-за создания отдельного файла для класса.



Defining an Anonymous Class

Анонимный класс - это специализированная форма локального класса, у которого нет имени. Он объявляется и создается в одном операторе с использованием ключевого слова new, имени типа в круглых скобках и набора фигурных скобок {}. Анонимные классы необходимы для расширения существующего класса или реализации существующего интерфейса. Они полезны, когда у вас есть небольшая реализация, которая больше нигде не будет использоваться.



Обратите особое внимание на точку с запятой. Мы объявляем локальную переменную в этих строках. Объявления локальных переменных должны заканчиваться точкой с запятой, как и другие операторы Java, даже если они длинные и содержат анонимный класс.

Understanding Garbage Collection

Как JVM узнает, подходит ли объект для сборки мусора? JVM терпеливо ожидает и отслеживает каждый объект, пока не определит, что коду больше не нужна эта память. Объект будет оставаться в куче до тех пор, пока он не станет недоступным. Объект становится недоступным при возникновении одной из двух ситуаций:

* У объекта больше нет ссылок, указывающих на него.
* Все ссылки на объект out of scope.