

Управление доступом

(Соккрытие реализации)

- ▶ Рефакторинг - переработка способного кода, направленная на то, чтобы сделать его более понятным. Необходимость в определении постоянной составляющей для удобства программистов-клиентов
- ▶ Спецификаторы доступа - указатель доступности `public`, `private`, `package-private(default)`, `protected`
- ▶ Пакет как библиотечный модуль содержит группу классов, объединенных в одном пространстве имен
- ▶ Файл с исходным текстом - компилируемый модуль. Имя каждого заканчивается на `.java` внутри `public` класс. Любой модуль может содержать не более одного открытого класса, остальные не публичные, а вспомогательные
- ▶ Для каждого класса `.java` в результате компиляции - ряд файлов с расширением `.class` - рабочая программа состоит из таких файлов, объединяются пакет и сжимаются в джар
- ▶ Библиотека - набор файлов с классами. В каждом файле имеется `public`-класс с не-`public` классами. Для объединения всех этих компонентов - используется ключевое слово `package`. Для использования классов в пакетах необходимо использовать ключевое слово `import *имя пакета*`

- ▶ Создание уникальных имен пакетов
 - ▶ Размещение файлов пакета в отдельном каталоге -> создание уникальных имен пакетов и обнаружение классов, путем использования иерархии; преобразование имени пакета в каталог на диске компьютера.
 - ▶ Для файлов jar - записать имя файла jar в переменной classpath, не ограничиваясь указанием пути к месту его расположения.
-
- ▶ Конфликты имен
 - ▶ При наличии одинаковых имен класса в двух разных пакетах происходит конфликт имен и компилятор требует добавить точку к пакету который вы хотите использовать и указать класс там, вот.

- ▶ Спецификаторы доступа
- ▶ Располагаются перед определением членов класса. Если его нет, то используется пакетный уровень доступа
- ▶ Доступ в пределах пакета - член класса доступен для всех классов текущего пакета. Член класса доступен для всех остальных классов текущего пакета, но для классов за пределами - приватный
- ▶ PUBLIC
- ▶ При использовании ключевого слова `public` вы объявляете, что следующее объявления члена класса доступно для всех (для программистов-клиентов в частности). Но если вы написали функцию вне класса, то к ней пакетный доступ.
- ▶ При нахождении классов в одном каталоге и отсутствии явно заданного имени пакета `java` включает файлы в пакет по умолчанию
- ▶ PRIVATE
- ▶ Доступ к члену класса не предоставляется никому, кроме методов этого класса. Другие классы в пакете тоже ничего не могут. Позволяет менять члены класса не боясь, что это отразится где-то еще. Может помочь в сокрытии процесса создания объекта (запрет на использование конструктора. Создание объекта в коде-клиента путем вызова функции)
- ▶ Все вспомогательные методы, поля - лучше `private`

▶ PROTECTED

- ▶ Связано с понятием наследования. Если при реализации наследования класс родитель - из другого пакета, новый класс получает доступ только к открытым членам из исходного пакета
- ▶ Если надо предоставить доступ к наследнику но закрыть от остальных - протектед - определяет защищенные члены класса, доступ для всех наследников + предоставляет доступ для всех в пределах пакета

▶ Интерфейс и реализация

- ▶ Контроль над доступом - сокрытие реализации Помещение данных и методов в классы с сокрытием реализации - инкапсуляция. Результат - новый тип данных, обладающий характеристиками и поведением. Закрытый доступ для программиста клиента и для разделения интерфейса и реализации
- ▶ При реализации функций в классе сначала публичные, потом защищенные, потом приватные.
- ▶ Для управления доступом - спецификатор перед ключевым словом class
- ▶ В каждом модуле может существовать один открытый класс.
- ▶ При создании класса с доступом в пределах пакета его поля рекомендуется помечать как private. Класс с пакетным доступом используется только в своем пакете . Класс не может быть private\protected

- ▶ Чтобы сделать класс приватным надо объявить все его конструкторы приватными. Использовать подобное можно следующим образом - определить статический метод, в котором создается объект и который возвращает ссылку на него. Создать объект как статическую приватную переменную и давать доступ через функцию (на синглтон похоже)

- ▶ Причины к ограничению доступа - предотвращение использования клиентами внутренней реализации класса, возможность изменения внутренней реализации не влияющая на клиента
- ▶ Открытый интерфейс - то, что видит пользователь
- ▶ Управление доступом ориентировано на создатель-клиент