

СПб АУ ИОЦНТ РАН

Java 06

29.03.2016

## Замечания про SimpleFTP

- ▶ Расположение деклараций: константы, поля, конструкторы, методы (внутри групп сортировка по visibility)

## Замечания про SimpleFTP

- ▶ Расположение деклараций: константы, поля, конструкторы, методы (внутри групп сортировка по visibility)
- ▶ Альтернатива: private методы после первого места использования

## Замечания про SimpleFTP

- ▶ Расположение деклараций: константы, поля, конструкторы, методы (внутри групп сортировка по visibility)
- ▶ Альтернатива: private методы после первого места использования
- ▶ Оптимизируйте visibility
- ▶ Оптимизируйте статичность

## Замечания про SimpleFTP

- ▶ Расположение деклараций: константы, поля, конструкторы, методы (внутри групп сортировка по visibility)
- ▶ Альтернатива: private методы после первого места использования
- ▶ Оптимизируйте visibility
- ▶ Оптимизируйте статичность
- ▶ Названия методов в большинстве случаев должны иметь глагольную форму

## Замечания про SimpleFTP

- ▶ Расположение деклараций: константы, поля, конструкторы, методы (внутри групп сортировка по visibility)
- ▶ Альтернатива: private методы после первого места использования
- ▶ Оптимизируйте visibility
- ▶ Оптимизируйте статичность
- ▶ Названия методов в большинстве случаев должны иметь глагольную форму
- ▶ Проверяйте список и содержимое файлов перед коммитом

# Замечания про SimpleFTP

- ▶ Расположение деклараций: константы, поля, конструкторы, методы (внутри групп сортировка по visibility)
- ▶ Альтернатива: private методы после первого места использования
- ▶ Оптимизируйте visibility
- ▶ Оптимизируйте статичность
- ▶ Названия методов в большинстве случаев должны иметь глагольную форму
- ▶ Проверяйте список и содержимое файлов перед коммитом
- ▶ Игнорирование исключений

# Замечания про SimpleFTP

- ▶ Расположение деклараций: константы, поля, конструкторы, методы (внутри групп сортировка по visibility)
- ▶ Альтернатива: private методы после первого места использования
- ▶ Оптимизируйте visibility
- ▶ Оптимизируйте статичность
- ▶ Названия методов в большинстве случаев должны иметь глагольную форму
- ▶ Проверяйте список и содержимое файлов перед коммитом
- ▶ Игнорирование исключений
- ▶ Писать о том, что вы сдали можно только после зеленых тестов (*настройте travis в своих репозиториях*)



# Замечания про SimpleFTP

- ▶ Расположение деклараций: константы, поля, конструкторы, методы (внутри групп сортировка по visibility)
- ▶ Альтернатива: private методы после первого места использования
- ▶ Оптимизируйте visibility
- ▶ Оптимизируйте статичность
- ▶ Названия методов в большинстве случаев должны иметь глагольную форму
- ▶ Проверяйте список и содержимое файлов перед коммитом
- ▶ Игнорирование исключений
- ▶ Писать о том, что вы сдали можно только после зеленых тестов (*настройте travis в своих репозиториях*)
- ▶ Подсказки IDE / компилятора

<https://github.com/EnterpriseQualityCoding/FizzBuzzEnterpriseEdition>

<https://github.com/EnterpriseQualityCoding/FizzBuzzEnterpriseEdition>

Когда стоит создавать классы

- ▶ Сложная предметная область с нетривиальными связями между объектами

<https://github.com/EnterpriseQualityCoding/FizzBuzzEnterpriseEdition>

Когда стоит создавать классы

- ▶ Сложная предметная область с нетривиальными связями между объектами
- ▶ Часто необходимо оперировать совокупностью данных из нескольких компонент (FileEntry)

<https://github.com/EnterpriseQualityCoding/FizzBuzzEnterpriseEdition>

Когда стоит создавать классы

- ▶ Сложная предметная область с нетривиальными связями между объектами
- ▶ Часто необходимо оперировать совокупностью данных из нескольких компонент (FileEntry)
- ▶ Реализация сложного абстрактного класса (много методов)

<https://github.com/EnterpriseQualityCoding/FizzBuzzEnterpriseEdition>

Когда стоит создавать классы

- ▶ Сложная предметная область с нетривиальными связями между объектами
- ▶ Часто необходимо оперировать совокупностью данных из нескольких компонент (FileEntry)
- ▶ Реализация сложного абстрактного класса (много методов)
- ▶ Хранилище утилитных static-методов

Когда стоит создавать интерфейсы? У вас естественным образом появилась абстракция в предметной области, которая:

- ▶ Является ключевой в вашем приложении

Когда стоит создавать интерфейсы? У вас естественным образом появилась абстракция в предметной области, которая:

- ▶ Является ключевой в вашем приложении
- ▶ Имеет нетривиальную реализацию



Когда стоит создавать интерфейсы? У вас естественным образом появилась абстракция в предметной области, которая:

- ▶ Является ключевой в вашем приложении
- ▶ Имеет нетривиальную реализацию
- ▶ Или несколько разных реализаций

Когда стоит создавать интерфейсы? У вас естественным образом появилась абстракция в предметной области, которая:

- ▶ Является ключевой в вашем приложении
- ▶ Имеет нетривиальную реализацию
- ▶ Или несколько разных реализаций
- ▶ Используется во многих уголках кода (причем без знаний о реализации)

Когда стоит создавать интерфейсы? У вас естественным образом появилась абстракция в предметной области, которая:

- ▶ Является ключевой в вашем приложении
- ▶ Имеет нетривиальную реализацию
- ▶ Или несколько разных реализаций
- ▶ Используется во многих уголках кода (причем без знаний о реализации)

Пожалуйста, не надо дублировать имеющийся в JDK набор интерфейсов: `Supplier`, `Consumer`, `Function` и т.д.

Когда стоит создавать интерфейсы? У вас естественным образом появилась абстракция в предметной области, которая:

- ▶ Является ключевой в вашем приложении
- ▶ Имеет нетривиальную реализацию
- ▶ Или несколько разных реализаций
- ▶ Используется во многих уголках кода (причем без знаний о реализации)

Пожалуйста, не надо дублировать имеющийся в JDK набор интерфейсов: `Supplier`, `Consumer`, `Function` и т.д.

Будьте внимательнее при сдаче работы про торренты, она будет оцениваться строже, чем обычно

## Задание. Реализация простого неблокирующего сервера

- ▶ Сервер сразу начинает отдавать клиенту содержимое одного файла, после чего разрывает соединение
- ▶ Содержимое файла можно перед началом работы прочитать в память
- ▶ Работа с клиентскими сокетами и открытие новых должно происходить в неблокирующем режиме
- ▶ Сервер должен работать в одном потоке
- ▶ Рекомендуется для клиентских сокетов включать `tcpNoDelay`

Дедлайн: 11.04 23:59