**Примерный план и материалы семинара 5:**

1. **Разобрать примеры**:

* Примеры (1…7) из пакета **sample\_interface\_impl.** К этим примерам можно потом вернуться: следующая лекция будет посвящена *reflection*, после чего будет разговор про лямбды (и тогда можно еще раз вспомнить пример 7 из этой серии примеров).Обращать внимание на комментарии и пояснения в тексте.

1. **Разобраться, как в IDEA можно собирать исполняемый JAR для Java-приложения и как его запускать из командной строки на исполнение.** См. инструкцию СборкаJARа.doc
2. **Проделать упражнение по созданию приложения с простым консольным диалогом** (для подготовки к контрольному домашнему заданию).

Требуется разработать консольное приложение, которое:

* в консоли приглашает пользователя ввести целое число;
* проверяет, что введено целое число (**int**); иначе выдается сообщение об ошибке и новое приглашение к вводу числа;
* выдает сообщение о четности или нечетности введенного числа;
* завершает работу при вводе команды “**exit**” (case-insensitive).

Проект должен:

* собираться и работать в IDEA; (показать/проверить);
* компилироваться и запускаться из командной строки (можно командными файлами); (показать/проверить);
* **собирать средствами IDEA** исполняемый **OddEvenApp.jar** , который запускается из командной строки: **java –jar OddEvenApp.jar**

**Те, кто не сделает на семинаре, выкладывают архив на google disk и присылают ссылку для скачивания (на проверку).**

1. **Прорешать вопросники** (из папки docs, можно выборочно).
2. **Следующие примеры можно оставить для домашнего рассмотрения:**

* sample\_1: использование static и default методов в интерфейсе;

полезно понаставить breakpoints повсюду и показать в debug-mode последовательность выполнения;

полезно открыть MyInterface.class декомпилятором и указать, как происходит «inlining» статически известных констант (на примере VALUE = 7);

* sample\_2: реализация и использование интерфейсов с одинаковыми методами (default и non-default);
* sample\_3: statically known final constants inline usage and initialization with inheritance;
* sample\_4: statically known constants in inner class static fields (that “is not allowed”); and changes in JDK16(+) – now they are allowed…(сравнить JDK 8 и 17)
* sample\_5: accessing outer class variables from inner class;
* sample\_6: пример необходимости final | effectively final в анонимных классах.