**План**

1. **Разобрать примеры:**
   * из подпакета **thread**s:

* **Test0** – thread’ы выполняются параллельно друг с другом недетерминированным образом;
* **Test1\_interrupt** – пример прерывания постоянно активного thread’а;
* **Test2\_Interrupt** – пример прерывания thread’ов, которые могут быть в ожидании по времени;
* **Test3\_Interrupt** – пример взаимодействия thread’ов (нарисовать диаграмму последовательности);
* **Test4\_JoinTest** - пример использования метода join();
* **Test5\_LettersTest** – пример использования ключевого слова ***synchronized***.
* **Test6\_Counter** – этот пример можно дать на 20 минут для самостоятельной работы (упражнение на **wait/notify**):
* показать, что простая реализация класса Storage не позволяет получать в консоль упорядоченные по возрастанью значения, и просто объявления synchronized – методов доступа к значению не помогают это сделать;
* надо реализовать класс Storage так, чтобы в консоль печатался натуральный ряд чисел. Ничего, кроме самого класса Storage, не изменять.
* Возможное решение – commented out… (см.)
  + из подпакета **jmm**:
* **VolatileSample –** пример использования ключевого слова ***volatile*;**
* **VolatileTests -** примеры взаимодействия thread’ов с помощью общей переменной (volatile и/или не volatile) – см. VolatileTest1;

аналогично – пример использования synchronized – доступа к общей переменной – см. VolatileTest2.

* + из подпакета **tricky**:
* объяснить, почему пример **Main1** работает верно, а пример **Main2** – нет...

1. **Прорешать Quiz:** Multithreading.doc