**Захаров Денис. М18-Ш01. Билет 27**

**1) Дайте определение понятию “поток-демон”.**

Поток-демон — это поток, выполняемый в фоне. Он выполняется вместе с программой, однако не является ее неотъемлемой частью. Суть потока-демона в обслуживании основного процесса. Примерами могут служить потоки таймера, посылающие другим потокам сигналы через определенные моменты времени, или потоки, очищающие кэш. Потоки-демоны не следует использовать для прикладных задач, таких как запись в базу данных, обращение к файлу и т.п., т.к эти потоки могут быть завершены в любой момент. В Java процесс завершается тогда, когда завершается последний его поток. Даже если метод main() уже завершился, но еще выполняются порожденные им потоки, система будет ждать их завершения.  
Однако это правило не относится к особому виду потоков – демонам. Если завершился последний обычный поток процесса, и остались только потоки-демоны, то они будут принудительно завершены и выполнение процесса закончится.

**2) Как создать поток-демон?**

Чтобы объявит поток демоном нужно перед запуском потока вызвать его метод setDaemon(true);  
Проверить, является ли поток демоном, можно вызвав его метод boolean isDaemon();

|  |  |
| --- | --- |
| 1  2  3 | Thread thread = new Thread(new DaemonClass());  thread.setDaemon(true);  System.out.println(thread.isDaemon()); //true |

**3)** **Как принудительно остановить поток?**

В Java 8 нет метода, который бы принудительно останавливал поток. Он может остановиться только сам. Java имеет встроенный механизм оповещения потока, который называется Interruption (прерывание, вмешательство).

**4) Что такое ThreadPoolExecutor и зачем он нужен?**

ThreadPoolExecutor состоит из фиксированного числа основных потоков, которые все время находятся внутри, и нескольких избыточных потоков, которые могут порождаться и затем прерываться, когда они больше не нужны. Параметр corePoolSize - это количество основных потоков, которые будут созданы и сохранены в пуле. Если все основные потоки заняты и передано больше задач, то пулу разрешается расти до maximumPoolSize.ThreadPoolExecutor выполняет переданную задачу (Callable или Runnable), используя один из внутренних доступных потоков из пула.

Если количество одновременно выполняемых задач всегда меньше или равно размеру пула, то они выполняются сразу же. В противном случае некоторые из этих задач могут быть помещены в очередь, чтобы дождаться своей очереди.

ThreadPoolExicutor позволяет выполнять задачи одновременно и уменьшать время работы программы.