***32.Свойства JSR1. Типы памяти. Типы нитей. Особенности компиляции.***

*JSR1 Real-Time Java*

Real time Java is a catch-all term for a combination of technologies that enables programmers to write programs that meet the demands of real-time systems in the Java programming language.

Java's sophisticated memory management, native support for threading and concurrency, type safety, and relative simplicity have created a demand for its use in many domains. Its capabilities have been enhanced to support real time computational needs:

* Real time Java supports a strict priority based threading model,
* because Java threads support priorities, Java locking mechanisms support priority inversion avoidance techniques, such as priority inheritance or the priority ceiling protocol, and
* event handling.

The Real-Time Specification for Java (RTSJ) is a set of interfaces and behavioral refinements that enable real-time computer programming in the Java programming language. RTSJ 1.0 was developed as JSR 1 under the Java Community Process, which approved the new standard in November, 2001.

* Реализации: Sun Java RTS, IBM WebSphere Real Time, Apogee Aphelion, TimeSys RTSJ
* Планирование
  + Невытесняющее планирование
  + Строгая семантика приоритетов
  + Наследование приоритетов
* Управление памятью
  + Бессмертная память (дальше есть про это)
  + Контекстная память
* Нити
  + Нити реального времени без кучи
  + Нити реального времени
  + Крайние сроки
* Синхронизация
  + Неждущие (lock free, wait free) очереди
* Время и таймеры
  + Периодические события
  + Часы повышенной точности
* Асинхронная обработка событий
  + Обработка периодических событий
  + Обработка системных сигналов
  + Обработка пропущенных крайних сроков
  + Обработка пользовательских событий

*Типы памяти*

* Бессмертная память
  + Класс ImmortalMemory [(javadoc)](http://jrate.sourceforge.net/api/stable/javax/realtime/ImmortalMemory.html)
  + Служит для статического выделения бессмертных разделяемых объектов
* Контекстная память
  + Класс ScopedMemory [(javadoc)](http://jrate.sourceforge.net/api/stable/javax/realtime/ScopedMemory.html)
  + Служит для выделения объектов с ограниченным временем жизни без использования сборки мусора
  + Контексты образуют иерархию
  + Для освобождения контекста используется подсчет ссылок
* Обычная куча
  + Использует сборку мусора

*Возможные ссылки*

*Permit = разрешить, permit = запретить*



*Типы нитей*

* Нить реального времени без кучи
  + Класс NoHeapRealtimeThread [(javadoc)](http://docs.oracle.com/javase/realtime/doc_2.2/release/rtsj-docs/javax/realtime/NoHeapRealtimeThread.html)
  + Может использовать только контекстную и бессмертную память
  + Не останавливается для сборки мусора
  + Не вытесняется при планировании
  + Обеспечивает наилучшее время реакции
  + Используется для обработки событий с коротким жестким крайним сроком
  + Сложности при программировании
* RealtimeThread
  + Класс RealTimeThread [(javadoc)](http://docs.oracle.com/javase/realtime/doc_2.2/release/rtsj-docs/javax/realtime/RealtimeThread.html)
  + Не вытесняется при планировании
  + Останавливается для сборки мусора
  + Время реакции определяется свойствами сборки мусора
  + Используется для обработки периодических событий с ограниченным крайним сроком
  + Использование мало отличается от обычных нитей
* Обычные нити
  + Класс Thread
  + Вытесняется при планировании
  + Останавливается для сборки мусора
  + Для нетребовательных к времени реакции задач

*Особенности компиляции*

* JIT компиляция
  + Ресурсы на компиляцию
  + Изменение производительности с течением времени
  + Требуется прогрев
* AOT компиляция
  + Невозможность межпроцедурной оптимизации
  + Усложнение развертывания
  + Постоянная производительность