

Developing Real-Time Applications for the Java™ Platform

Course Number: DTJ-4103

5-day instructor-led training with labs

Price in USD: \$3,000 (call for Team Training price)

To register, please call: 1-800-422-8020

COURSE DESCRIPTION

The Developing Real-Time Java™ Programming Applications course provides students with the essential skills required to use the real-time capabilities of the Real-Time Specification for Java (RTSJ) to create JSR-01 compliant real-time applications.

WHO CAN BENEFIT

Students who can benefit from this course are programmers, developers, and software architects who are involved in development of real-time systems using the Java programming language with the RTSJ extensions. Students should be fluent in the Java programming language and should be familiar with the nature of real-time programming, though extensive prior experience is not required.

SKILLS GAINED

Upon completion of this course, students should be able to:

- Control garbage collection induced jitter in thread release
- Create real-time thread objects, and use scheduling parameters and periodic parameters to control thread behavior
- Control concurrent access to resources and describe priority inversion and the priority inheritance algorithm
- Understand the constraints that apply when using objects in non-heap memory, and issues that might arise when using library APIs in non-heap memory
- Use AsyncEvent, OneShotTimer, PeriodicTimer, and AsyncEventHandler objects to provide timed and asynchronous behavior
- Control the queuing of asynchronous events to limit the impact of out-of-spec events or handlers on other parts of a system
- Interrupt threads asynchronously and control their behavior under such conditions
- Use portals to access scoped memory
- Describe and use appropriately the scoped run loop, encapsulated method, handoff, timed algorithm, multi-scoped object, wedge thread pattern
- Use overrun handlers to limit the impact of out-of-spec threads on other parts of a system

COURSE OUTLINE

- **Introduction to the Real-Time Java System (RTS)***
 - Describe real-time programming and the Java RTS
 - Outline the relationships between IO, resource contention, garbage collection and real-time systems
 - Outline memory types provided by RTSJ
 - Describe fundamental real-time architectures
 - Describe and design a timeline system
 - Create a periodic real-time thread
 - Measure the latency and jitter of a periodic RT thread
- **Interacting with the Real World**
 - Design objects for use in a real-time system
 - Describe actuators and sensors
 - Describe use of JNI, RMA, polling, and happenings for communicating with IO devices
- **Creating the Timeline Executive**
 - Determine time constraints affecting tasks
 - Determine a base frequency for a timeline architecture and assign tasks to timeslots
 - Implement a timeline architecture
 - Describe how to determine frame costs
 - Implement cost overrun handlers
 - Describe benefits and costs of the timeline architecture

continued on back

- **Applying the Scoped Run Loop Pattern to the Timeline Architecture**
 - Describe how the garbage collected heap can introduce jitter in real time response
 - Describe how scoped and immortal memory spaces can avoid GC effects
 - Describe how and when space allocated to objects in different memory types is recovered
 - State the basic rules limiting references between memory types
 - Describe and implement the scoped run-loop pattern
 - Recognize situations that might cause jitter or non-determinism in real-time threads even when the heap is not directly referenced
 - Configure and use initialization-time compilation
- **Introduction to Event Driven Architecture**
 - Describe the essential elements of an event-driven system, and the interactions between them
 - List benefits and costs of an event-driven approach
 - Describe how a single CPU is scheduled when multiple threads of differing priorities are runnable
 - Design sound objects for use in an event-driven system
 - Describe the RTSJ priority scheduler
 - Write code for active components in RTSJ
- **Data Synchronization for Concurrent Tasks**
 - Identify shared resources in a system
 - Select appropriate arbitration mechanisms for concurrent resource access
 - Recognize critical sections and protect them with synchronized blocks
 - Describe priority inversion and priority inversion avoidance techniques
- **Handling Asynchronous Events**
 - Describe events and happenings in the RTSJ
 - Use the classes AsyncEvent and AsyncEventHandler
 - Distinguish bound and unbound event handlers

- Write code to handle a termination handler for a program
- Write code to create and handle one-shot and periodic timer events and application
- Write code to issue an event under program control
- Describe arrival time queues and write code to control arrival time queue overflow behavior
- **Scoped Memory Issues and Patterns**
 - Describe the single-parent rule and a thread's active scope stack
 - Outline problems that arise when using library APIs in non-heap memory spaces
 - Describe and implement the encapsulated method pattern
 - Determine the necessary size for a memory scope
- **Component State Patterns**
 - Identify and differentiate component lifetimes and choose appropriate approaches for handling these
 - Describe and implement the multi-scoped object pattern
 - Describe and use the portal object of a ScopedMemory object
 - Describe concurrency approaches suited to portals
 - Describe and implement the wedge thread pattern
 - Describe approaches for moving data between scopes
 - Describe and implement the handoff pattern
- **Asynchronous Transfer of Control**
 - Describe and implement the timed algorithm pattern
 - Describe and use the asynchronous transfer of control APIs
 - Write interruptible code, and code that defers interrupts when necessary
- **Pipeline Architecture**
 - Describe the pipeline architecture
 - Describe and use wait-free queues
 - List benefits and costs of the pipeline architecture
 - List benefits and costs of a distributed pipeline architecture

For further information on courses for developers please visit our web site at sun.com/training/catalog/java/index.xml

Specifications are subject to change without notice.

*For further information on the Java Real-Time System product, please visit our website at java.sun.com/j2se/realtime