Feedback — Week 4 Quiz

Help Center

You submitted this quiz on **Wed 15 Apr 2015 8:43 PM PDT**. You got a score of **48.00** out of **48.00**.

Question 1

Which of the following are Gang-of-Four patterns applied to the Android concurrency frameworks, according to the videos in week #4

Your Answer		Score	Explanation
☐ Monitor Object	✔	1.00	
☐ Visitor	✔	1.00	
☐ Active Object	✔	1.00	
☑ Template Method	✔	1.00	
☑ Strategy	✔	1.00	
☑ Factory Method	✔	1.00	
Total		6.00 / 6.00	

Question Explanation

Please see the week #4 video on "Overview of Patterns in the Android Concurrency Frameworks"

Question 2

Which of the following are POSA patterns applied in the Android concurrency frameworks, according to the videos in week #3

Your Answer	Score	Explanation

☑ Thread-Specific Storage	✔	1.00
☑ Half-Sync/Half-Async	✔	1.00
☐ Template Method	✔	1.00
☐ Bridge	✔	1.00
☑ Command Processor	✔	1.00
☐ Extension Interface	✔	1.00
Total		6.00 / 6.00

Question Explanation

Please see the week #4 video on "Overview of Patterns in the Android Concurrency Frameworks"

Question 3

Which of the following patterns are applied in the corresponding frameworks, according to the videos in week #4

Your Answer		Score	Explanation
☐ The Thread-Specific Storage pattern is applied in the Java Executor framework	✔	1.00	
☑ The Factory Method pattern is applied in the AsyncTask framework	✔	1.00	
☐ The Strategy pattern is applied in the HaMeR framework	✔	1.00	
☑ The Command Processor pattern is applied in the HaMeR framework	✔	1.00	
☑ The Active Object pattern is applied in the Java Executor framework	✔	1.00	
☐ The Bridge pattern is applied in the HaMeR framework	✔	1.00	
Total		6.00 / 6.00	

Question Explanation

Please see the week #4 video on "Overview of Patterns in the Android Concurrency Frameworks"

Question 4

Which of the following are positive consequences (i.e., benefits) of applying the Thread-Specific Storage pattern, according to the videos in week #4

Your Answer		Score	Explanation
☐ It defers some steps in its concurrent processing algorithm to a subclass	✔	1.00	
☑ It elimiinates snchronization overhead for accesses to an object	✔	1.00	
☐ It allows subclasses to override hook methods	✔	1.00	
☐ It decouples interface from implementation so the two can vary independently	✔	1.00	
Total		4.00 / 4.00	

Question Explanation

Please see the week #4 video on "The Thread-Specific Storage pattern"

Question 5

Which of the following are known uses of the Thread-Specific Storage pattern according to the videos in week #4

Your Answer		Score	Explanation
☑ The Android Looper class	✔	1.00	
☐ The Android IntentService	✔	1.00	

☑ The standard C errno macro	✔	1.00
☐ The Java Executor framework	✔	1.00
Total		4.00 / 4.00

Question Explanation

Please see the week #4 video on "The Thread-Specific Storage pattern"

Question 6

Which of the following is the intent of the Command Processor pattern, according to the videos in week #4

Your Answer		Score	Explanation
☐ Allow multiple threads to use one "logically global" access point without incurring locking overhead on each object access	✔	1.00	
☐ Decouple asynchronous (async) and synchronous (sync) service processing in a concurrent system by introducing two intercommunicating layers--one for async and one for sync service processing--to simplify programming without unduly reducing performance	✔	1.00	
☑ Package a piece of application functionality--as well as its parameterization in an object--to make it usable in another context	✔	1.00	
☐ Define service requests on components as the units of concurrency and run service requests on a component in different thread(s) from the requesting client thread	✔	1.00	
Total		4.00 / 4.00	

Question Explanation

Please see the week #4 video on "The Command Processor pattern"

Question 7

Which of the following are negative consequences (drawbacks) of the Command Processor pattern, according to the videos in week #4

Your Answer		Score	Explanation
☐ The client isn't blocked for duration of the processing	✔	1.00	
☐ Supports context- and time-independent execution of the application logic	✔	1.00	
☑ Functionality is driven by the sender rather than the receiver	✔	1.00	
☑ Supporting two-way operations requires additional programming effort	✔	1.00	
Total		4.00 / 4.00	

Question Explanation

Please see the week #4 video on "The Command Processor pattern"

Question 8

Which of the following are examples of situations under which the Active Object pattern should be applied, as shown in the videos from week #4

Your Answer		Score	Explanation
☑ When multiple client method requests can run concurrently on an object	✔	1.20	
☐ When there's a need to decouple the decision of what piece of code should be executed from the decision of what context or time this should happen	✔	1.20	
☐ When it's necessary to retrofit legacy code to be thread-safe	✔	1.20	
☑ When an object's interface methods should define its	✔	1.20	

concurrency boundaries		
☑ When an object's methods may block for a long duration during their execution	✔	1.20
☑ When additional capabilities must be implemented consistently for all requests to a service	✖	0.00
Total		6.00 / 6.00

Question Explanation

Please see the week #4 video on "The Active Object pattern"

Question 9

Which of the following are known uses of the Active Object pattern, according to the video from week #4

Your Answer		Score	Explanation
☑ The Android HaMeR framework	✔	1.00	
☐ The Android IntentService framework	✔	1.00	
☑ The Java ExecutorService framework	✔	1.00	
☐ The Java Executor framework	✔	1.00	
Total		4.00 / 4.00	

Question Explanation

Please see the week #4 video on "The Active Object pattern"

Question 10

Which of the following best describe the motivation for the Half-Sync/Half-Async pattern, according to the videos in week #3

Your Answer		Score	Explanation
☐ Ensure only one Looper resides within each thread in a concurrent program	✔	1.00	
☑ Simplify concurrent programming without unduly reducing performance	✔	1.00	
☐ Ensure enhancements to the services offered by a component don't break existing code	✔	1.00	
☑ Separate short-duration from long-duration operations in concurrent programs	✔	1.00	
Total		4.00 / 4.00	

Question Explanation

Please see the week #4 video on "The Half-Sync/Half-Async pattern"