**Примерные оценочные материалы, применяемые при проведении промежуточной аттестации по дисциплине (модулю)**

**«Проектирование мобильных приложений»**

## Примерный перечень практических заданий

1. Спроектировать одноэкранное мобильное приложение с использованием архитектуры MVP, Kotlin Coroutines для обеспечения асинхронных запросов в сеть и с адаптивным пользовательским интерфейсом на основе Android Jetpack.

2. Спроектировать одноэкранное мобильное приложение с использованием архитектуры MVVM, Kotlin Coroutines для обеспечения асинхронных запросов в сеть и с адаптивным пользовательским интерфейсом на основе Android Jetpack.

3. Спроектировать одноэкранное мобильное приложение с использованием архитектуры MVI, Kotlin Coroutines для обеспечения асинхронных запросов в сеть и с адаптивным пользовательским интерфейсом на основе Android Jetpack.

4. Спроектировать одноэкранное мобильное приложение с использованием архитектуры VIPER, Kotlin Coroutines для обеспечения асинхронных запросов в сеть и с адаптивным пользовательским интерфейсом на основе Android Jetpack.

5. Спроектировать одноэкранное мобильное приложение с использованием архитектуры VIP, Kotlin Coroutines для обеспечения асинхронных запросов в сеть и с адаптивным пользовательским интерфейсом на основе Android Jetpack.

6. Спроектировать одноэкранное мобильное приложение с использованием архитектуры Redux, Kotlin Coroutines для обеспечения асинхронных запросов в сеть и с адаптивным пользовательским интерфейсом на основе Android Jetpack.

7. Спроектировать одноэкранное мобильное приложение с использованием архитектуры RIB, Kotlin Coroutines для обеспечения асинхронных запросов в сеть и с адаптивным пользовательским интерфейсом на основе Android Jetpack.

8. Спроектировать одноэкранное мобильное приложение с использованием архитектуры Elm, Kotlin Coroutines для обеспечения асинхронных запросов в сеть и с адаптивным пользовательским интерфейсом на основе Android Jetpack.

9. Спроектировать одноэкранное мобильное приложение с использованием архитектуры TCA, Kotlin Coroutines для обеспечения асинхронных запросов в сеть и с адаптивным пользовательским интерфейсом на основе Android Jetpack.

10. Спроектировать многоэкранное мобильное приложение с использованием архитектуры MVP, Kotlin Coroutines для обеспечения асинхронных запросов в сеть и с адаптивным пользовательским интерфейсом на основе Android Jetpack.

11. Спроектировать многоэкранное мобильное приложение с использованием архитектуры MVVM, Kotlin Coroutines для обеспечения асинхронных запросов в сеть и с адаптивным пользовательским интерфейсом на основе Android Jetpack. 12. Спроектировать многоэкранное мобильное приложение с использованием архитектуры MVI, Kotlin Coroutines для обеспечения асинхронных запросов в сеть и с адаптивным пользовательским интерфейсом на основе Android Jetpack.

13. Спроектировать многоэкранное мобильное приложение с использованием архитектуры VIPER, Kotlin Coroutines для обеспечения асинхронных запросов в сеть и с адаптивным пользовательским интерфейсом на основе Android Jetpack.

14. Спроектировать многоэкранное мобильное приложение с использованием архитектуры VIP, Kotlin Coroutines для обеспечения асинхронных запросов в сеть и с адаптивным пользовательским интерфейсом на основе Android Jetpack.

15. Спроектировать многоэкранное мобильное приложение с использованием архитектуры Redux, Kotlin Coroutines для обеспечения асинхронных запросов в сеть и с адаптивным пользовательским интерфейсом на основе Android Jetpack.

16. Спроектировать многоэкранное мобильное приложение с использованием архитектуры Redux, Kotlin Coroutines для обеспечения асинхронных запросов в сеть и с адаптивным пользовательским интерфейсом на основе Android Jetpack.

17. Спроектировать многоэкранное мобильное приложение с использованием архитектуры Elm, Kotlin Coroutines для обеспечения асинхронных запросов в сеть и с адаптивным пользовательским интерфейсом на основе Android Jetpack.

18. Спроектировать многоэкранное мобильное приложение с использованием архитектуры TCA, Kotlin Coroutines для обеспечения асинхронных запросов в сеть и с адаптивным пользовательским интерфейсом на основе Android Jetpack.

**Критерии оценивания практических заданий**

Уровень знаний определяется оценками «отлично», «хорошо», «удовлетворительно», «неудовлетворительно».

Оценка «отлично» - студент показывает полные и глубокие знания программного материала, логично и аргументировано отвечает на поставленный вопрос, а также дополнительные вопросы, показывает высокий уровень теоретических и практических знаний.

Оценка «хорошо» - студент показывает глубокие знания программного материала, грамотно его излагает, достаточно полно отвечает на поставленный вопрос и дополнительные вопросы, умело формулирует выводы. В тоже время при ответе допускает несущественные погрешности.

Оценка «удовлетворительно» - студент показывает достаточные, но не глубокие знания программного материала; при ответе не допускает грубых ошибок или противоречий, однако в формулировании ответа отсутствует должная связь между анализом, аргументацией и выводами. Для получения правильного ответа требуется уточняющие вопросы.

Оценка «неудовлетворительно» - студент показывает недостаточные знания программного материала, не способен аргументировано и последовательно его излагать, допускает грубые ошибки в ответах, неправильно отвечает на поставленный вопрос или затрудняется с ответом.