

КПО. Модуль 2. Темы к коллоквиуму

Формат проведения

Коллоквиум пройдет в режиме собеседования. Каждому студенту выдается по три небольших теоретических вопроса по каждой из трех тем. Необходимо сразу дать ответы без подготовки. Преподаватель может задать дополнительные вопросы по каждой из тем. На основе полученных ответов формируется три промежуточных оценки:

O_T1 - оценка за устный ответ по теме 1.

O_T2 - оценка за устный ответ по теме 2.

O_T3 - оценка за устный ответ по теме 3.

Итоговая оценка за коллоквиум:

$O_{KP2} = (O_{T1} + O_{T2} + O_{T3}) / 3$.

Максимальная оценка 8 баллов.

Тема 1. Основы языка UML

1.1. Назначение UML.

Что такое UML и каково его основное назначение?

1.2. Основные виды диаграмм UML

Какие основные типы диаграмм используются в UML? Объясните, для чего эти диаграммы применяются.

1.3. Диаграммы UML

Какой функционал необходимо описать при моделировании системы заказов доставки посылок?

1.4. Диаграммы классов UML

Кратко опишите, какие классы необходимы для моделирования системы заказов посылок? Какие связи необходимы между классами?

1.5. Диаграммы последовательностей UML

Кратко опишите диаграмму последовательностей для прецедента авторизации пользователя.

1.6. Диаграммы активностей UML

Кратко опишите диаграмму активностей для прецедента авторизации пользователя.

Тема 2. Основы языка программирования Kotlin

2.1. Классы в Kotlin

Зачем нужны классы? Как создаются классы в Kotlin? Что такое блок `init`? Что такое `primary constructor`? В каком порядке они вызываются?

2.2. Свойства и поля в Kotlin

Какими способами можно создать свойство в Kotlin? Зачем классам нужны свойства?

2.3. Наследование в Kotlin

Зачем применяют наследование? Как наследуются интерфейсы и классы в Kotlin?

2.4. Основные структуры данных в Kotlin

Какие основные структуры данных имеются в Kotlin. Зачем они нужны?

2.5. Null-безопасность в Kotlin

Что такое null-безопасность в Kotlin? Зачем она нужна?

2.6. Code style в Kotlin

Виды написания составных слов. В каких случаях используется тот или иной тип?

2.7. Лямбда-выражения в Kotlin

Что такое лямбда-выражение? Зачему они нужны? Что такое функция высшего порядка? Где в Kotlin они используются?

Тема 3. Принципы и концепции программирования

3.1. Принципы SOLID

Что представляют собой принципы SOLID в контексте программирования?

3.2. Принципы KISS и DRY

Объясните принципы KISS и DRY. Для чего сформулированы эти принципы?

3.3. Основные парадигмы программирования.

Каковы основные парадигмы программирования и в чем их отличительные особенности?

3.4. Основы ООП.

Что такое ООП? Устно приведите пример реализации каждого принципа на Kotlin.
Каковы основные достоинства и недостатки этой парадигмы программирования?

Удачи!