Бауманский учебный центр «Специалист»

Программа курса Scala 3: Классы и объектно - ориентированное программирование

Код курса: **СКАЛА2**

Тема	Ак. часов
Модуль 1. Введение в объектно ориентированное программирование на Scala 3 • Зачем нам ООП? • Необходимость моделирования предметной области. • Избавление от дублирования кода. • Реализация однотипных элементов в Scala 3. • Обзор связей между объектами и классами в ООП.	**1**
Модуль 2. Создание классов и объектов • Данные и методы класса. • Ссылка this и её использование. • Основной и вспомогательный конструктор. • Инкапсуляции. • Реализация паттерна одиночка. **Практика:** • Проектирование и написание класса. • Реализация инкапсуляции. • Добавление в класс удобных/полезных методов, например, +(сложение), apply, update, foreach и т.д.	**5**
Модуль 3. Наследование в Scala 3 • Проектирование иерархий. • Основной конструктор при наследовании. • Ссылка super и её использование. • Модификатор доступа protected • Запрет/разрешение наследования. • Рефлексия и метаданные. **Практика:** • Иерархия геометрических фигур (часть 1).	**4**
Модуль 4. Полиморфизм в Scala 3 • Абстрактные классы. • Преобразование/выбор типа ссылки в иерархиях. • Полиморфизм. **Практика:** • Иерархия геометрических фигур (часть 2).	**4**

Модуль 5. Иерархия классов Scala 3 • Класс Any и его методы. • Классы AnyVal и AnyRef. • Класс Unit и Null. • Класс Nothing. **Практика:** • Пример наследования от класса AnyVal.	**1.5**
Модуль 6. Изоляция имён в Scala 3 • Области видимость имён. • Использование пакетов для изоляции имён. • Импортирование имён из других пакетов. **Практика:** • Написать свой класс Any и реализовать в нем toString, == и hashCode.	**1.5**
Модуль 7. Вспомогательные типы в Scala 3 • Case class. • Case object. • Перечисления (enum). • Соответствие шаблону. **Практика:** • Использовать коллекцию (Map) для хранения фигур. • Изменить реализацию на case class.	**2**
Модуль 8. Scala Traits • Особенности типа данных Trait. • Зачем нам Traits? **Практика:** Сортировка массива объектов. • Линеаризация наследования trait в Scala. • Динамическое поведение super у trait. • Многократное подмешивание (stackable modifications) поведения при помощи trait. **Практика:** • Добавление нового поведения в классы фигур с использованием Scala Traits.	**5**
Аудиторная нагрузка в классе с преподавателем	**24 ак.ч. + 24 ак.ч. самостоятельно**

По окончании обучения на курсе проводится итоговая аттестация. Аттестация проводится в виде теста на последнем занятии или на основании оценок практических работ, выполняемых во время обучения на курсе.

Контакты центра

Для частных лиц
+7 (495) 232-32-16

Для организаций
+7 (495) 780-48-44

E-mail
info@specialist.ru