

Лабораторная работа №3
по дисциплине
«Функциональное программирование»

Введение в функциональное программирование и Scala

Тема: Pattern matching и case-классы

Цель работы:

Изучить механизм сопоставления с образцом (pattern matching).
Научиться использовать case-классы и алгебраические типы данных (ADT).
Реализовать простые структуры данных и их обработку.

Теоретические основы:

- Синтаксис pattern matching в Scala
- Case-классы, их особенности и преимущества
- Алгебраические типы данных (ADT)
- Полиморфизм и типизация

Задания:

1. Создать case-класс с несколькими полями и использовать его в pattern matching.
2. Реализовать функцию, которая принимает значение произвольного типа и обрабатывает его с помощью pattern matching.
3. Создать ADT (например, тип для арифметических выражений) и написать функцию вычисления значения выражения.
4. Продемонстрировать работу полиморфной функции с использованием дженериков.

Контрольные вопросы

1. Назовите ключевые принципы функционального программирования. Чем они отличаются от императивного подхода?
2. Что такое «чистые функции»? Приведите пример в Scala.
3. Объясните термин «иммутабельность». Почему она важна в ФП?
4. В чем разница между `val` и `var`? Когда что использовать?
5. Что такое `case`-классы? Какие преимущества они дают?
6. Как работает `pattern matching` в Scala? Приведите пример.
7. Что такое функция высшего порядка? Приведите пример из стандартной библиотеки Scala.
8. Объясните, как работают `map`, `filter` и `reduce` для коллекций.
9. Что такое каррирование? Как оно реализуется в Scala?
10. Что такое алгебраические типы данных (ADT)? Приведите пример.