

## Аннотация рабочей программы дисциплины

**Направление подготовки:** 02.03.02 «Фундаментальная информатика и информационные технологии»

**Тип образовательной программы:** академический бакалавриат

**Направленность (профиль):** Информационная сфера

**Квалификация (степень) выпускника:** бакалавр

**Форма обучения:** очная

### 1. Наименование дисциплины

Б1.В.ДВ.8.1 Современные технологии программирования

### 2. Цели и задачи дисциплины (модуля):

Целями освоения дисциплины «Современные технологии разработки» являются ознакомление студентов с современными технологиями разработки, приемами решения практических задач, формирование практических умений и навыков, необходимых для приобретения квалификации бакалавра прикладной информатики. Дисциплина «Современные технологии разработки» проводится в форме лабораторных, практических работ и самостоятельной работы студента.

Задачи дисциплины: дать специальные знания по дисциплине, достичь достаточного уровня знаний по современным технологиям программирования и сформировать у студентов практические навыки использования современных технологий программирования.

### 3. Требования к результатам освоения дисциплины (модуля):

Процесс изучения дисциплины направлен на формирование следующих компетенций: ОПК-2, ОПК-3, ПК-5.

В результате изучения дисциплины студент должен:

**Знать:** базовые алгоритмы обработки информации; методы функционального программирования.

**Уметь:** формализовать поставленную задачу; составлять и оформлять программы на языках программирования для виртуальной машины Java; тестировать и отлаживать программы в интегрированной среде разработки; применять полученные знания к различным предметным областям.

**Владеть:** навыками разработки, отладки и тестирования программ в интегрированной среде разработки, практическими навыками решения прикладных задач.

### 4. Объем дисциплины (модуля) и виды учебной работы

Вид учебной работы	Всего часов / зачетных единиц	Семестры
		7
Аудиторные занятия (всего)	66	66
Практические занятия (ПЗ)	30	30
Лабораторные работы (ЛР)	30	30
Контроль самостоятельной работы (КСР)	6	6
<b>Самостоятельная работа (всего)</b>	6	6
Вид промежуточной аттестации		Зачет
Общая трудоемкость часы	72	72
зачетные единицы	2	2

### 5. Краткая характеристика содержания учебной дисциплины

## РАЗДЕЛ 1. НОВЫЕ ВОЗМОЖНОСТИ ЯЗЫКА ПРОГРАММИРОВАНИЯ JAVA 8

### **ТЕМА 1.1. Лямбда-выражения**

Функция как параметр. Лямбда-выражения. Выведение типов.

### **ТЕМА 1.2. Функциональные интерфейсы**

Аннотация `@Override`. Аннотация `@FunctionalInterface`. Ссылки на методы.

### **ТЕМА 1.3. Пакет `java.util.function`**

Пакет `java.util.function`. Типизированные и обобщенные интерфейсы. Интерфейсы `java.util.function`. Общие случаи. Интерфейс `Predicate`. Метод `firstMatch`. Интерфейс `Function`. Интерфейс `BinaryOperator`. Интерфейс `Consumer`. Интерфейс `Supplier`. Область видимости переменных. Вызов метода экземпляра класса.

### **ТЕМА 1.4. Потoki для работы с коллекциями**

Поток. Сравнение потоков и списков. Особенности потоков. Преобразование в поток и обратно. Методы потоков: `foreach`, `reduce`. Ограничение потока. Проверка соответствия. Параллельные потоки. Бесконечные потоки.

## **РАЗДЕЛ 2. ЯЗЫК ПРОГРАММИРОВАНИЯ SCALA**

### **ТЕМА 2.1. Основы Scala. Управляющие структуры и функции**

Получение и установка Scala. Требования к оформлению кода на Scala. Интерактивный интерпретатор (REPL). Объявление переменных. Типы данных. Арифметика и перегрузка операторов. Вызов функций и методов. Метод `apply`. Scaladoc. Условные выражения. Завершение инструкций. Блочные выражения. Ввод и вывод. Циклы. Расширенный цикл `for`. Функции и процедуры. Отложенные вычисления. Исключения

### **ТЕМА 2.2. Функциональное программирование**

Функции как значение. Анонимные функции. Функции с функциональными параметрами. Вывод типов. Замыкания. Карринг. Абстракция управляющих конструкций. Выражение `return`.

### **ТЕМА 2.3. Массивы, ассоциативные массивы и кортежи**

Массивы фиксированной длины. Массивы переменной длины. Обход массивов. Преобразование массивов. Типичные алгоритмы. Многомерные массивы. Ассоциативные массивы. Кортежи.

### **ТЕМА 2.4. Объектно-ориентированное программирование**

Классы и методы без параметров. Дополнительные конструкторы. Главный конструктор. Вложенные классы. Объекты-одиночки. Объекты-компаньоны. Объекты, расширяющие классы. Метод `apply`. Объект, представляющий приложение. Перечисления.

### **ТЕМА 2.5. Наследование**

Наследование классов. Переопределение методов. Создание суперклассов. Анонимные подклассы. Абстрактные классы. Иерархия наследования в Scala.

### **ТЕМА 2.6. Коллекции. Сопоставление с образцом**

Основные трейты коллекций. Изменяемые и неизменяемые коллекции. Последовательности. Списки. Множества. Добавление и удаление элементов. Общие методы. Функции `map` и `flatMap`. Итераторы. Сопоставление с образцом, отличия от `switch`. Ограничители. Переменные в образцах. Сопоставление с типами, массивами, списками, кортежами. Экстракторы. Case-классы

### **ТЕМА 2.7. Трейты**

Отсутствие множественного наследования. Трейты как интерфейсы. Трейты с конкретными реализациями. Объекты с трейтами. Многоуровневые трейты. Порядок конструирования трейтов.

## **6. Форма промежуточной аттестации:**

зачет

## **7. Разработчик аннотации**

Доцент кафедры алгебраических и информационных систем А.С. Зинченко