



Java SE 8 Программирование/ Java SE 8 Programming

Код курса: JSE8P

Java SE 8 Программирование/ Java SE 8 Programming

Код курса: JSE8P

Длительность	40 ак. часов
Формат	Очно; Дистанционно
Разработчик курса	Oracle
Тип	Учебный курс
Способ обучения	Под руководством тренера

О курсе

В курсе рассматриваются ключевые возможности языка и программные интерфейсы (API), которые используются для создания объектно-ориентированных приложений на базе платформы Java Standard Edition (SE) 8.

Подробная информация

Профиль аудитории:

Разработчики.

Предварительные требования:

- Знания в объёме курса «Основы программирования на Java SE 8».

По окончании курса слушатели смогут:

- Создавать высокопроизводительные многопоточные приложения;
- Создавать приложения Java с использованием объектно-ориентированных возможностей языка, таких как инкапсуляция, наследование и полиморфизм;
- Реализовывать функционал ввода-вывода для чтения и записи данных в файлы;
- Выполнять приложения Java из командной строки;
- Работать с файлами, каталогами и файловыми системами с использованием спецификаций JDK NIO.2;
- Создавать приложения с использованием Java Collections Framework;
- Выполнять различные операции над таблицами баз данных, такие как создание, чтение, обновление и удаление, с использованием технологий JDBC и JPA;
- Осуществлять поиск и фильтрацию в коллекциях с использованием выражений Lambda;
- Использовать техники обработки ошибок с использованием обработки исключений;
- Использовать функции конкурентности (concurrency) выражений Lambda.

Программа курса

Модуль 1 «Обзор платформы Java»

- Цели и задачи курса
- Программы Java - платформенно-независимые
- Группы продуктов Java Technology
- Версии платформы Java SE
- Загрузка и установка Java SDK
- Java в серверной среде
- Интернет вещей
- Сообщество Java
- Java Community Process (JCP)
- OpenJDK
- Поддержка Oracle Java SE
- Дополнительные ресурсы

Модуль 2 «Синтаксис Java и обзор классов»

- Обзор языка Java
- Структура классов Java
- Правила наименования в Java
- Компиляция и запуск
- Блоки кода
- Простые (Primitive) типы данных
- Numeric Literals
- Операторы
- Логические операторы
- Выражение IF Else
- Выражение Switch
- Цикл While
- Цикл Do-While
- Цикл For
- Массивы и цикл For-Each
- Строковые значения
- Операции над строковыми значениями: StringBuilder
- Пример простого класса Java: Employee
- Методы
- Создание экземпляра класса
- Конструкторы
- Выражение Package
- Выражения Imports
- Java is Pass-By-Value
- Объекты, передаваемые в качестве параметров
- Garbage Collection
- Лабораторная работа «Создание классов Java»

Модуль 3 «Инкапсуляция и подклассы»

- Инкапсуляция
- Пример инкапсуляции
- Инкапсуляция: Публичные и частные модификаторы доступа (Access Modifiers)
- Инкапсуляция: Частные данные, публичные методы
- Пример: модифицированный класс Employee
- Неизменные (immutable) классы
- Наименования методов: рекомендации
- Преимущества инкапсуляции
- Создание подклассов
- Подклассы
- Manager Subclass
- Конструкторы в подклассах
- Использование super
- Перегрузка (overloading) методов
- Перегруженные конструкторы
- Перегруженные конструкторы: пример
- Единичное (single) наследование
- Лабораторная работа «Создание подклассов»

Модуль 4 «Переопределение методов, полиморфизм и статические классы»

- Контроль доступа
- Защищенный контроль доступа: пример
- Рекомендации по использованию контроля доступа
- Переопределение методов
- Вызов виртуального метода
- Доступность переопределенных методов
- Применение полиморфизма
- Использование ключевого слова (keyword) InstanceOf
- Переопределение объектных методов (Object Method)
- Объектный метод toString
- Объектный метод equals
- Переопределение equals в Employee
- Переопределение объекта hashCode
- Методы, использующие переменные аргументы (Variable Arguments)
- Распределение (Casting) ссылок на объекты (Object references)
- Правила распределения Upward
- Правила распределения Downward
- Ключевое слово Static
- Статические методы
- Использование статических переменных и методов: пример
- Применение статических методов
- Статические переменные
- Определение статических переменных
- Использование статических переменных
- Статические инициализаторы (Initializers)
- Статический импорт
- Шаблоны дизайна (Design Patterns)
- Паттерн Singleton

- Пример использования Singleton
- Неизменные классы
- Пример создания неизменного класса в Java
- Лабораторная работа «Переопределение методов и применение полиморфизма»
- Лабораторная работа «Применение шаблона дизайна Singleton»

Модуль 5 «Абстрактные и вложенные классы»

- Моделирование бизнес-проблем с использованием классов
- Включение обобщения (generalization)
- Необходимость использования абстрактных классов
- Определение абстрактных классов
- Определение абстрактных методов
- Проверка абстрактных классов
- Финальные (Final) методы
- Финальные (Final) классы
- Финальные (Final) переменные
- Декларация финальных переменных
- Вложенные классы
- Пример: класс Member
- Перечисление (Enumeration)
- Использование перечисления
- Комплексные перечисления
- Лабораторная работа «Применение ключевого слова Abstract»
- Лабораторная работа «Использование внутреннего класса в качестве вспомогательного»
- Лабораторная работа «Использование перечислений»

Модуль 6 «Интерфейсы и выражения Lambda»

- Интерфейсы Java
- Проблемы, решаемые интерфейсами
- Класс CrashedRock
- Интерфейс SalesCalc
- Добавление интерфейса
- Ссылки к интерфейсам
- Применимость ссылок к интерфейсам
- Гибкость кода интерфейсов
- Методы default в интерфейсах
- Пример: метод default
- Методы static в интерфейсах
- Поля Constraint
- Расширение интерфейсов
- Применение расширения
- Анонимные внутренние классы
- Пример анонимных внутренних классов
- Регулярный класс: String Analysis
- Пример использования String Analysis
- Интерфейс String Analysis
- Инкапсуляция цикла for Loop

- Класс String Analysis с методом Helper
- Лямбда-выражение String Analysis
- Определенное (defined) лямбда-выражение
- Что такое лямбда-выражение?
- Лямбда-выражение Shorthand
- Лямбда-выражение в качестве переменной
- Лабораторная работа «Применение интерфейса»
- Лабораторная работа «Использование интерфейсов Java»
- Лабораторная работа «Создание лямбда-выражения»

Модуль 7 «Коллекции и обобщения (дженерики)»

- Обобщения
- Простой класс Cache без обобщений
- Обобщенный класс Cache
- Обобщения в действии
- Обобщения с типом интерфейса Diamond
- Коллекции
- Типы коллекций
- Коллекции интерфейсов и их применение
- Интерфейс List
- ArrayList
- Autoboxing и Unboxing
- ArrayList без обобщений
- Обобщенный ArrayList
- Обобщенный ArrayList: Iteration и Boxing
- Интерфейс Set
- TreeSet: применение Set
- Интерфейс Map
- Типы Map
- TreeMap: применение Map
- Интерфейс Deque
- Пример: Stack with Deque
- Упорядоченные (ordering) коллекции
- Пример Comparable
- Интерфейс Comparator
- Пример Comparator
- Лабораторная работа «Подсчет Part Numbers с использованием HashMap»
- Лабораторная работа «Реализация Stack с использованием объекта Deque»

Модуль 8 «Коллекции, потоки и фильтры»

- Коллекции, потоки и фильтры
- Класс Person
- Свойства Person
- Паттерн Builder
- Коллекция Iteration и лямбды
- RobocallTest07: поток и фильтр
- RobocallTest07: поток и снова фильтр

- Класс SalesTxn
- Потоки Java
- Метод Filter
- Ссылки метода
- Связывание (chaining) методов
- Определение конвейера (Pipeline Defined)
- Лабораторная работа «Коллекции, потоки и фильтры»

Модуль 9 «Встроенные функциональные интерфейсы лямбда»

- Встроенные функциональные интерфейсы
- Пакет util.function
- Примеры допущений (assumptions)
- Предикаты
- Примеры предикатов
- Consumer
- Пример Consumer
- Функция
- Пример функции
- Supplier
- Пример Supplier
- Примитивный интерфейс
- Возвращение примитивного типа
- Возвращение примитивного типа: пример
- Бинарные типы
- Бинарные типы: пример
- Оператор Unary
- Оператор Unary: пример
- Обзор обобщений WildCard
- Лабораторная работа «Встроенные функциональные интерфейсы лямбда»

Модуль 10 «Операции лямбда»

- Stream API
- Типы операций
- Извлечение данных с использованием Map
- Обзор методов Search
- Методы Search
- Опциональный класс
- «Ленивые» (lazy) операции
- Методы Stream Data
- Выполнение вычислений
- Сортировка
- Обновления Comparator
- Сохранение данных с использованием Stream
- Класс Collector
- Быстрые потоки с использованием Stream.of
- Выравнивание (flatten) данных с использованием flatMap
- Лабораторная работа «Операции лямбда»

Модуль 11 «Исключения (exceptions) и утверждения (assertions)»

- Обработка ошибок
- Обработка ошибок в Java
- Выражение try-catch
- Объекты исключений
- Категории исключений
- Условие finally
- Перехват множественных исключений
- Декларация исключений
- Обработка декларированных исключений
- Throwing Exceptions
- Модифицированные (custom) исключения
- Допущения
- Синтаксис допущений
- Внутренние инварианты
- Контроль потока инвариантов
- Класс инвариантов
- Контроль оценки допущений в процессе выполнения
- Лабораторная работа «Перехват исключений»
- Лабораторная работа «Расширение исключений: использование throw и throws»

Модуль 12 «Java Date/Time API»

- Почему дата и время важны?
- Дата и время в предыдущих версиях
- Java Date/Time API: цели
- Работа с локальными датой и временем
- Работа с LocalDate
- Работа с LocalTime
- Работа с LocalDateTime
- Работа с временными зонами
- Правила перехода на летнее время
- Моделирование временных зон
- Создание объектов ZonedDateTime
- Работа с разрывами (gap) и перекрытиями (overlap) в ZonedDateTime
- ZoneRules
- Работа с разными временными зонами
- Методы даты и времени
- Количественные значения (amounts) даты и времени
- Период
- Длительность
- Калькуляции между днями
- Оформление дат
- Использование нотации Fluent
- Лабораторная работа «Работа с датами и временем»

Модуль 13 «Основы Java I/O»

- Основы Java I/O
- Потоки I/O
- Приложения I/O
- Данные внутри потока
- Методы Byte Stream InputStream
- Методы Byte Stream OutputStream
- Методы чтения Character Stream
- Методы записи Character Stream
- Связывание (chaining) потоков I/O
- Console I/O
- Запись в стандартный вывод
- Чтение из стандартного ввода
- Channel I/O
- Постоянство (Persistence)
- Сериализация и графы объектов
- Переходные (transient) поля и объекты
- Serial Version UID
- Запись и чтение Object Stream
- Методы сериализации
- Лабораторная работа «Создание простого консольного I/O приложения»
- Лабораторная работа «Сериализация и десериализация ShoppingCart»

Модуль 14 «Java File I/O (NIO.2)»

- New File I/O API (NIO.2)
- Ограничения io.file
- Файловые системы, пути, файлы
- Относительный и абсолютный пути
- Концепции 2
- Интерфейс Path
- Возможности интерфейса Path
- Создание Subpath
- Объединение двух Paths
- Символьные ссылки
- Работа со ссылками
- Операции с файлами
- Проверка файла или каталога
- Создание файла или каталога
- Удаление файла или каталога
- Копирование файла или каталога
- Перемещение файла или каталога
- Вывод содержимого каталога
- Просмотр структуры каталога
- Файловый поток BufferedReader
- Файловый поток NIO
- Чтение файла в ArrayList
- Управление метаданными
- Символические ссылки
- Лабораторная работа «Работа с файлами и каталогами»

Модуль 15 «Конкурентность (concurrency)»

- Планирование задач
- Устаревшие Thread и Runnable
- Расширение Thread
- Применение Runnable
- Пакет util.concurrent
- Рекомендованные классы Threading
- util.ExecutorService
- Остановка util.ExecutorService
- util.concurrent.Callable
- util.concurrent.Future
- Sharded-данные
- Проблемы с Sharded-данными
- Nonharded-данные
- Атомарные операции
- Выполнение Out-of-Order
- Ключевое слово synchronized
- Методы synchronized
- Блоки synchronized
- Object Monitor Locking
- Производительность Threading
- Классы и пакеты Java.util.concurrent
- Пакет util.concurrent.atomic
- util.concurrent.CyclicBarrier
- Коллекции ThreadSafe
- Лабораторная работа «Использование пакета Java.util.concurrent»

Модуль 16 «Фреймворк Fork-Join»

- Параллелизм
- Отсутствие параллелизма
- Naïve параллелизм
- Необходимость во фреймворке Fork-Join
- Work-Stealing
- util.concurrent.ForkJoinTask
- Структура compute
- Рекомендации по работе с фреймворком Fork-Join
- Лабораторная работа «Использование фреймворка Fork-Join»

Модуль 17 «Параллельные потоки»

- Обзор потоков
- Обработка коллекций в старом стиле
- Обработка коллекций в новом стиле
- Конвейер потоков
- Сравнение стилей
- Параллельный поток
- Использование параллельных потоков: Коллекции

- Использование параллельных потоков: Из потока
- Конвейеры
- Embrace Statefulness
- Детерминистские потоки
- Недетерминистские потоки
- Сокращение
- Пример параллельного выполнения
- Производительность
- Простая модель производительности
- Лабораторная работа «Параллельные потоки»

Модуль 18 «Создание приложений баз данных с использованием JDBC»

- Использование JDBC API
- Использование класса Vendor's Driver
- Ключевые компоненты JDBC API
- Написание запросов и получение результатов
- Использование объекта ResultSet
- Операции CRUD с использованием JDBC API: Извлечение
- Операции CRUD с использованием JDBC: Извлечение
- Операции CRUD с использованием JDBC API: Создание
- Операции CRUD с использованием JDBC API: Обновление
- Операции CRUD с использованием JDBC API: Удаление
- Класс SQLException
- Заккрытие объектов JDBC
- Конструкт try-with-resources
- Использование PreparedStatement
- Использование PreparedStatement: настройка параметров
- Выполнение PreparedStatement
- PreparedStatement: Использование цикла для установки значений
- Использование CallableStatement
- Лабораторная работа «Работа с базой данных Derby и JDBC»

Модуль 19 «Локализация»

- Почему необходима локализация?
- Регион (locale)
- Свойства
- Загрузка и использование файла свойств
- Загрузка свойств из командной строки
- Комплект (bundle) ресурсов
- Файл комплекта ресурсов
- Примеры файлов комплекта ресурсов
- Инициализация простого приложения
- Простое приложение: Main Loop
- Метод printMenu
- Изменение региона
- Простой интерфейс на французском языке
- Формат даты и валюты

- Отображение валюты
- Форматирование валюты с использованием NumberFormat
- Отображение дат
- Отображение дат с использованием DateTimeFormatter
- Стили форматирования
- Лабораторная работа «Создание локализованного приложения»

[Посмотреть расписание курса и записаться на обучение](#)

Обращайтесь по любым вопросам
к менеджерам Учебного центра Softline

8 (800) 505-05-07 | edusales@softline.com

Ждём вас на занятиях в Учебном центре Softline!



Почему Учебный центр Softline?

Лидер на рынке корпоративного обучения.

Более 300 тысяч подготовленных IT-специалистов.

Гибкий индивидуальный подход в обучении, скидки и акции.

Широкая сеть представительств в крупнейших городах РФ и СНГ; дистанционный формат обучение на вашей территории или в арендованном классе в любой точке мира.

Высокотехнологичное оборудование

Более **18 лет** опыта работы

Международные сертификаты для IT-специалистов и пользователей в Центрах тестирования

Сертифицированные тренеры с богатым практическим опытом работы

Авторизации от мировых производителей ПО (Microsoft, Cisco, VMware, Citrix, Лаборатория Касперского, Oracle, Autodesk, Код безопасности и других).

Разработка курсов и тестов под заказ, внедрение корпоративных систем обучения.

Подробнее об Учебном центре Softline

Вы можете узнать из [профайла](#).