



ИТ-ОБРАЗОВАНИЕ В ПЕТЕРБУРГЕ И УДАЛЕННО

[Очно](#) [Онлайн](#) [Корпоративное обучение](#) [Индивидуальное обучение](#) [Часто задаваемые вопросы \(FAQ\)](#)

[Главная](#) / [Курсы программирования с нуля](#) / [Программирование](#)

Курс .NET- разработчик (джун -> мидл)

.NET

Начало курса	29.10.2024
Продолжительность	3 месяца (80 ак. часов)
Расписание	Вт 19:00 - 21:30 Пт 19:00 - 21:30

Онлайн
(10 мест)

78990 Р

Единовременно

ЗАПИСАТЬСЯ

7899 Р

В рассрочку / 10 мес

ЗАПИСАТЬСЯ

Также доступна рассрочка на 4, 6 и 10 месяцев.

Доступные
способы оплаты



Описание курса

Курс ".NET- разработчик (джун -> мидл)" предназначен для разработчиков, имеющих опыт разработки на любом современном языке программирования и желающих познакомиться с платформой .NET и языком C#.

Учащиеся пройдут путь от основ платформы и языка до подходов и инструментов решения типовых задач, включая разработку сложных веб-приложений.

Упор в курсе сделан на разработку backend-сервисов в современной парадигме "чистой" архитектуры и CQRS. Отдельно в рамках курса будут рассмотрены вопросы взаимодействия сервисов с хранилищами данных (БД, кеш), провайдерами аутентификации и очередями сообщений.

Практика:



- сквозная практическая работа по проекту – инкрементальное развитие сложной системы по предоставленной спецификации
- упор на работу с кодом в среде разработки – в виде презентаций подаётся самый минимум информации

Основные темы курса:

- базовый синтаксис языка C# (структура программы, языковые конструкции и операторы)
- система типов
- коллекции и структуры данных
- реализация ООП в C#
- Base Class Library (BCL)
- Language-integrated Queries (LINQ)
- делегаты и события
- основы платформы .NET (компиляция, сборка мусора, CLR, кросс-платформенные возможности)
- асинхронное и многопоточное программирование
- работа с хранилищами данных (БД, кеш)
- устройство и основные возможности ASP.NET Core
- реализация "чистой" архитектуры и CQRS
- межсервисная интеграция
- сервисные логи и метрики
- юнит- и интеграционное тестирование в .NET

Требования к участникам курса:

- понимание основных концепций любого языка программирования – переменные, операторы, циклы, коллекции и структуры данных
- знание основ ООП (полиморфизм, наследование, инкапсуляция, паттерны)
- знание основных подходов к работе с кодом (SOLID, DRY, KISS, рефакторинг)
- понимание основ юнит- и интеграционного тестирования
- базовые навыки работы с терминалом (запуск приложений с параметрами, навигация по файловой системе)
- умение работать с современными IDE (Visual Studio, VS Code, JetBrains IntelliJIdea/Rider)
- умение работать с системой контроля версий (git) и оболочками поверх неё (GitHub, Gitlab, Bitbucket)
- навыки базовой работы с Docker (образы, контейнеры, docker-compose)
- знание основ SQL и теории БД (реляционные, NoSQL, CAP-теорема)
- настроенная и готовая к работе среда Visual Studio Community Edition 2022 (или Professional)
- настроенный и готовый к работе локальный инстанс Docker с запущенным актуальным (latest) образом PostgreSQL

Знакомство с языком C# и платформой .NET



- создание простого консольного приложения
- структура решения (solution)
- метод Main
- работа с консольным выводом
- объявление и инициализация переменных
- примитивные типы
- простые арифметические операции
- параметры метода Main, передача параметров при запуске консольного приложения
- top level statements
- добавление проекта в решение. Сборка. Зависимости

Управление потоком выполнения. Операторы



- условные операторы
- оператор switch
- арифметические операторы
- логические операторы
- операторы управления потоком выполнения
- циклы
- рекурсия

Работа с массивами



- объявление и инициализация массива
- обход массива
- N-мерные массивы
- jagged-массивы
- обращение по индексу
- поиск элемента в массиве
- вставка и удаление элемента массива
- конкатенация массивов

Система типов



- тип object
- ссылочные типы. Куча
- значимые типы. Стек
- упаковка и распаковка объектов значимых типов
- явное приведение типов
- неявное приведение типов
- проверка на тип. Операторы is/as
- пределы допустимых значений для арифметических типов

Классы



- объявление класса. Сигнатура класса
- области видимости типов
- поля
- методы. Сигнатура метода. Параметры метода. Области видимости методов
- ковариантность и контравариантность методов
- свойства
- ключевые слова readonly и const
- семантика передачи по ссылке
- ключевые слова in, ref и out
- нейминг классов и его членов
- статические классы и члены класса
- extension-методы

Наследование



- понятие и назначение наследования в C#
- базовый класс. Объявление наследников
- наследование членов базового класса
- переопределение членов базового класса. Ключевое слово virtual/override
- ограничение наследования класса и членов класса. Ключевое слово sealed
- сокрытие членов базового класса. Ключевое слово new
- таблица виртуальных методов
- абстрактные классы
- проблема множественного наследования

- понятие интерфейса
- объявление интерфейса
- дефолтная реализация методов в интерфейсах
- приведение к типу интерфейса
- решение проблемы множественного наследования с помощью интерфейсов
- явная реализация метода интерфейса
- выбор между абстрактным классом и интерфейсом

Записи



- понятие и назначение записи
- объявление записи
- неизменяемость записей
- эквивалентность записей
- primary constructor
- создание копии записи с помощью конструкции with

Значимые типы данных



- понятие и назначение структур
- наследование в структурах
- семантика передачи по значению
- понятие и назначение перечислений. Битовые флаги
- readonly struct
- record struct

Работа со строками



- неизменяемость строк
- конкатенация и интерполяция
- специальные символы
- интернирование строк
- класс StringBuilder
- оптимизация работы со строками. Класс Span

- внутреннее устройство коллекций. Интерфейс IEnumerable
- generic-типы
- набор значений (HashSet)
- список (List)
- словарь (Dictionary<K,V>)
- связный список
- очередь (Queue)
- стек (Stack)
- ковариантность и контравариантность коллекций

LINQ



- назначение LINQ
- лямбда-выражения
- выборка
- фильтрация
- сортировка
- группировка
- отложенное выполнение вычислений. Замыкания

Делегаты и события



- понятие делегата
- определение делегата
- классы Func и Action
- цепочка делегатов
- method group
- понятие события
- определение события
- подписка на событие
- паттерн Observer

Кортежи и анонимные типы данных



- понятие и назначение анонимного типа
- объявление и использование анонимного типа
- понятие и назначение кортежа

Обработка исключений



- виды исключений
- устройство механизма обработки исключений в .NET
- конструкция try/catch
- блок finally
- работа с экземпляром исключения в блоке catch

Code First vs Database First



- Code First. Миграции
- Практикуемся с миграциями
- Database First. Scaffolding
- Практикуемся с созданием классов из существующей БД

Конкурентное программирование. Асинхронные операции



- понятие конкурентного программирования
- отличие асинхронного программирования от многопоточного
- понятие процесса и потока
- запуск процессов и потоков из кода
- управление потоками напрямую
- пул потоков
- класс Task. Создание задачи и управление её жизненным циклом
- класс TaskFactory
- async/await. Конечный автомат. Оповещение о завершении асинхронной операции
- объединение и отмена выполнения задач
- обработка исключений в Task

Многопоточное программирование



- Task Parallel Library. Класс Parallel
- PLINQ
- виды синхронизации конкурентного доступа
- основные примитивы синхронизации уровня приложения
- основные примитивы синхронизации уровня операционной системы
- гибридные примитивы синхронизации

Возможности платформы .NET



- устройство платформы .NET. Среда выполнения (CLR)
- основные возможности платформы
- жизненный цикл кода. Roslyn, IL, JIT
- сборка мусора. Garbage collector
- понятие неуправляемого ресурса
- детерминированная и недетерминированная финализация
- паттерн Disposable
- оператор using

Работа с хранилищами данных - интеграция с БД



- возможности интеграции кода с БД
- устройство и возможности EntityFramework
- подключение провайдера данных в коде приложения
- Code First vs DB First
- Code First. Подготовка контекста и моделей
- Code First. Миграции
- выполнение запроса на выборку
- выполнение вставки/обновления данных в БД
- отслеживание изменений
- интерфейс IQueryable
- анализ текста запроса, сгенерированного провайдером
- navigation properties. Отношения "one-to-one", "one-to-many", "many-to-many"
- производительность запросов

Кеширование



- понятие и назначение кеширования
- способы реализации кеширования в приложениях .NET
- in-memory кеш
- интеграция с внешним распределенным кешем

Фреймворк ASP.NET Core



- устройство и назначение ASP.NET Core

- контейнер управления зависимостями. Middleware
- ASP.NET Core Minimal API. Роутинг
- реализация HTTP-эндпоинтов в контроллерах
- подключение провайдера аутентификации
- настройка правил авторизации

Основы проектирования архитектуры сервиса ASP.NET Core



- развитие архитектурных подходов к проектированию сервисов
- понятие "чистой" архитектуры
- доменный слой
- слой инфраструктуры
- сервисный слой
- слой представления
- интерпретации "чистой" архитектуры
- проектирование слоёв в сервисе согласно "чистой" архитектуре
- основы CQRS. Команды, запросы, обработчики
- дедупликация логики в CQRS-подходе

Основы межсервисной интеграции



- способы интеграции сервисов
- интеграция посредством HTTP
- интеграция посредством брокера сообщений
- плюсы и минусы синхронного и асинхронного взаимодействия сервисов
- проблемы межсервисной интеграции. Надёжность, гарантии доставки
- архитектурные паттерны межсервисной интеграции

Логирование и сбор метрик сервисов



- назначение логов и метрик
- настройка провайдера логирования в коде сервиса
- настройка публикации метрик в коде сервиса
- анализ логов и метрик сторонними инструментами

Тестирование .NET-приложений



- проектирование юнит-тестов
- использование моков для внешних зависимостей
- проектирование интеграционных тестов
- организация сквозного процесса тестирования приложений

Презентация проектов. Подведение итогов курса



Документы после обучения

По окончании обучения вы получите сертификат. Нас знают и ценят работодатели, поэтому сертификат станет вашим дополнительным преимуществом при трудоустройстве в IT-компанию.

Сертификат о прохождении курса мы отправим вам на почту. Вы легко сможете добавить ссылку на сертификат в резюме и подтвердить свой уровень подготовки.

Преподаватели


[ЗАДАТЬ ВОПРОС](#)

Алексей

.NET Team Lead в **Тинькофф**

Закончил Самарский Государственный технический университет по специальности "Профессиональное обучение - Информатика, вычислительная техника, компьютерные технологии".

С самых ранних лет карьеры четко определил для себя, что тривиальная работа с 9 до 18 – это скучно. Это привело Алексея к участию в многочисленных конференциях DotNext (в том числе, в качестве спикера и организатора на стенде DotNetRu), созданию Самарского комьюнити .NET-разработчиков и обучению начинающих разработчиков в EPAM Lab.

Работал в группе компаний "Деловые Линии", Яндекс.Деньги, EPAM. В данное время трудится в должности .NET Team Lead во благо ведущей дистрибьюторской компании России – OCS Distribution.

Фото с очных занятий в СПб