## CS background + С# basics

1. Data structures and algorithms.
   1. Computational complexity. O(n) notation
   2. Sort/find algorithms, binary search
   3. Stacks, queues, linked lists, associative arrays, BST, hash-tables
   4. Recursions and Stack overflow
   5. [Optional] Trees/graphs, BSF and DSF traversal
   6. **Task:** Computational complexity for given practical task(e.g. find most common element in the List<int>): propose and compare few variants

Вопрос: Почему именно строки сделали неизменяемыми?

Вопрос: Как работает выделение памяти для строк в этих двух случаях:

1. string x = “111”;
2. string y = new string (‘1’, 3);

Альтернативный упрощенный вопрос: Каков механизм интернирования строк.

1. Design principles, SOLID, patterns, architecture styles
   1. Что такое ООП и как реализовано в C#
   2. SOLID
   3. Тут бы я не углублялся в теорию: общие понятия, примеры из практики с описанием проблемы и как решили в реальном проекте
   4. [Senior, optional] Event sourcing, CQRS, micro services, Client-server, Layer Architecture, Thin/Fat Client, MVC and other MV\*
   5. Unit and integration testing. Code coverage tools

Вопрос: Как разорвать циклические зависимости при Dependency Injection?

[Senior] Вопрос: CAP теорема, пример систем с разными сильными сочетаниями (CA, CP, AP) и их влияние на общую производительность.

1. C# basics
   1. Boxing/unboxing. Struct vs class.
   2. Dispose vs destructor
   3. virtual/override/new
   4. Stack memory vs heap memory. Any GC for stack objects?
   5. [Senior+] Why managed heap allocations are fast?
   6. [Senior+] GC. Mark and compact.
   7. **Task:** Имеем масив byte[90\_000], расскажите об особенностях работы GC с таким масивом. (проверяем знание по GC LOH и особенности его работы + негативный эффект на перфоманс в случае интенсивного выделения больших блоков памяти разной длины)
   8. Ask what makes sense: generics, properties, interfaces, pattern matching, attributes, delegate/event, static ctor, extension methods, using, yield, collections, etc -
   9. [Senior] How to implement plugin system in .Net.
   10. **Task**: What is the output of following code:

try {

try {

throw new Exception(“1”);

}finally {

throw new Exception(“2”);

}

}

catch (Exception ex) { Console.WrilteLine (ex.Message); }

1. Multithreading
   1. Sync primitives
   2. Mutex, semaphor, monitor, events, when not to use spinlock, interlocked operations. (
   3. Await vs Task.Result

Вопрос: Какие накладные расходы влияющие на производительность влечет за собой использование await/async

Вопрос: Что такое Synchronization Context, примеры контекста.

* 1. Double-checked locking
  2. Thread vs process
  3. IPC
  4. [Optional??] UI thread
  5. **Optional task:** thread-safe blocking queue with MaxCount limit

Вопрос: Как можно реализовать функцию «Такое приложение уже запущено» в Desktop приложении?

Вопрос: Почему и как могут возникнуть deadlocks в WPF UI приложениях при использовании await/async.

1. Networking
   1. TCP vs UDP basics
   2. REST
   3. [optional] HTTP basics. 1.0 vs 1.1 vs 2.0
   4. Based on candidate experience: sockets, WCF, Asp.Net (http modules/handlers, middleware), etc?
   5. **Optional task:** synchronous facade hiding asynchronous service

Вопрос: Как сделать пинг удаленного хоста стандартными средствами dotNet.

Вопрос: Что такое “Loopback”, особенности тестирования сетевого кода на локальной машине.

1. DB high-level overview
   1. CRUD understanding
   2. SQL vs noSQL
   3. Normalization/De-normalization understanding
   4. Transactions – что это, зачем нужны, какие проблемы решают, ACID, примеры уровней изоляции
2. [OPTIONAL] based on CV
   1. Docker, K8s, graphQL, Message queue, twelve-factor app, etc

WPF Specialization

1. What is MVVM?
2. What is Command Design Pattern and ICommand in WPF?
3. What is the Data Binding concept and How Binding works in WPF?
4. What are the Binding Modes in XAML?
5. What is Dependency property in WPF?
6. What is Attached property in WPF?
7. What are Resources in WPF (static and dynamic)?
8. What is the difference between Static and Dynamic resources?
9. What is value convertor in WPF?
10. What is Trigger and how many types of triggers in WPF?
11. What are the Templates (Control Template, Items Panel Template, and Data Template) in WPF?
12. What is a Routed event (Bubbling Events, Tunneling Events, and Direct Events)?
13. What difference between visual and logical WPF tree?

Вопрос: Как создать таблицу размерностью 1M (col) x 1M (row).

Вопрос: Как реализовать асинхронный биндинг.

Web Specialization (web development + SQL/noSQL)

1. Общие вопросы:
   1. Типы запросов: HttpGet, HttpPost, HttpPut, HttpDelete, HttpHead
   2. Как можно хранить данные между запросами? (HTTP Headers, Cookies, Session, Application, custom implementation... etc.)
   3. Как можно организовать обмен данными между клиентом и сервером? (HTTP requests, WebSockets, Long polling, SignalR, etc.)
   4. Как ограничить доступ к API методу? (anonymous access, authorized, JWT-токен, OAuth, ... etc.)
   5. Routing. Зачем это нужно и как настроить? (route constraints, routes ordering, routing matching)
   6. CORS (Cross Origin Resource Sharing).
   7. REST and RESTfull сервисы.
   8. Как можно тестировать API метод? (Swagger, Postman, Fiddler, load testing tools, xUnit, ... etc.)
   9. Как логировать application exceptions?
   10. ILogger, настройка логера (категории...), провайдеры логгирования.
   11. Dependency Injection: для чего нужно? Какие фреймворки знаете?
2. SQL specific (move to web specialization?) – OPTIONAL (в рамках вопроса ниже)
   1. GroupBy, Having keyword
   2. Normalization/De-normalization understanding
   3. Select query to several tables
   4. Stored procedures
   5. Transactions
   6. Indexes
   7. Triggers
   8. Views
   9. Functions
   10. Replication and partitioning
   11. Abstraction leak for ORM
   12. [Optional] Isolation levels, Jobs, Backups, security
   13. [Optional] Query optimizations, execution plan

Вопрос: Как сделать так, чтобы конкретный SQL запрос выполнялся не за 10 минут, а максимум за секунду? (в ходе ответа обращать внимание на вышеуказанные пункты)

1. ASP.NET
   1. HttpModule vs HttpHandler.
   2. Что такое ASP.NET State Service?
   3. ApiController vs Controller

## ASP.NET Core

1. ASP.NET Core Regular
   1. Startup, Назначение методов ConfigureServices, Configure
   2. Жизненный цикл middleware,
   3. Конвейер обработки запроса
   4. Методы Use, Run, Map, MapWhen
   5. Как добавить поддержку статик контента (UseStaticFiles, UseDefaultFiles)
   6. Передача конфигурации через Ioptions
   7. Может ли один и тот же API метод возвращать данные в разных форматах? Если так, то как запросить данные в XML формате? Custom I/O data formats (model binding, formatters)
2. ASP.NET Core Regular/Senior
   1. HttpClient middleware,
   2. Обработка исключений (UseDeveloperExceptionPage, UseExceptionHandler + Map, UseStatusCodePages, UseStatusCodePagesWithRedirects, web.config секция system.webServer/httpErrors)
   3. Dependency Injection, Services lifetime. Варианты инжекшена.
      * (Services lifetime: Transient, Scoped, Singleton, Регистрация: AddTransient, AddScoped, AddSingleton)
      * (Варианты инжекшена: параметры метода Startup.Configure, Конструкторы, HttpContext.RequestServices, app.ApplicationServices, Метод Invoke/InvokeAsync (middleware))
   4. Конфигурация, environment overrides.
   5. Проекция конфигурации на классы (Configuration.Bind(...); Configuration.Get<...>)
   6. Состояние приложения. Куки. Сессии (AddDistributedMemoryCache, AddSession+UseSession, HttpContext.Session.Keys, HttpContext.Items, HttpContext.Request.Cookies)
   7. ILogger, настройки категрии логера, фабрики логгера, провайдеры логгирования.
   8. Маршрутизация (UseEndpoints + MapXXX, RouterMiddleware RouteHandler+RouteBuilder+UseRouter)
   9. URL matching, Определение маршрутов, Ограничения маршрутов (IRouteConstraint)
   10. MVC, Сервисы MVC (AddMvc vs AddMvcCore, AddControllersWithViews, AddControllers), Контроллеры, Атрибуты NonController, ActionName и NonAction, типы запросов (HttpGet, HttpPost, HttpPut, HttpDelete, HttpHead)
   11. Представления, Razor render engine
   12. Аутентификация и авторизация (UseAuthentication, UseAuthorization), AuthorizeAttribute, AllowAnonymous,
   13. IPrincipal.IsInRole, IIdentity.IsAuthenticated
3. ASP.NET Core Senior
   1. Configuration providers.
   2. JWT-токен и его структура, Авторизация с помощью JWT-токенов (AddJwtBearer)
   3. SignalR Core (WebSockets, Server-Side Events, Long Polling) OPTIONAL
   4. SignalR.Hub создание и конфигурирование (AddSignalR, UseEndpoints + MapHub) OPTIONAL
   5. CORS и кросс-доменные запросы, Политики CORS (AddCors, UseCors, CorsPolicyBuilder)
   6. Работа с HTTPS (UseHttpsRedirection, SSL, HttpsRedirectionOptions)
   7. Что такое HSTS (HTTP Strict Transport Security Protocol) как его включить, зачем. (UseHsts, AddHsts, host excluding) OPTIONAL

Вопрос: Как организовать автоматическое обновление конфигурационных параметров в случае изменения XXXX.confg файла.

## Soft skills

1. Unclear requirements – understood differently by DEV and QA
2. Code review – reviewer requires to follow legacy approach
3. PROD defect (severity MAJOR)
4. Bonus: “floating” bug reproduced only in PROD  
   Bonus 2: dev team has no PROD access
5. Change request that does not fit into app architecture (or just looks strange)
6. Bonus: urgent, already signed off by the Client
7. Integration – bug in other system, workaround or not?
8. Bonus - work on same flow or codebase with another (Indian) vendor
9. Firm deadline – time that we have to implement is ½ of estimated