**Блок 1: Основы c#.**

1. Классы, структуры – как устроены, в чем отличие, когда что использовать. Разница в поведении Equals.
2. Оператор ref – как работает с классами и структурами
3. Boxing, unboxing – как работает
4. Строки – как устроены, в чем особенность.Интернирование
5. Когда и зачем используется StringBuilder.
6. Сборка мусора – как работает, когда и кем вызывается, какие есть оптимизации.
7. Методы Finalize и Dispose, в чем разница, когда что используется.
8. Управляемые и неуправляемые ресурсы
9. Потоки(Stream) – что это и как работает. Какие существуют(стримы источников, декорируюущие). Концепция стримов.
10. Делегаты – что такое.
11. События – как связаны с делегатами. Как устроены внутри? Lock free?
12. Лямбда выражения – что такое
13. IEnumerable – как работает внутри, IEnumerator
14. Yield return (switch case)
15. Типы коллекций когда и что применять (массив, список, хэш-таблица)
16. Как внутри устроена хэш-таблица
17. Связь Equals и GetHashCode
18. IAsyncEnumerable – если слышал, уже хорошо
19. Как решаются коллизии
20. LINQ – как устроен, как применяется, основные методы
21. Что такое поток в .net (Thread)
22. IO-bound, CPU-bound операции
23. Асинхронность – Task, как устроен, пул потоков, стэйт машина
24. Как работают async await (Обертка в ContinueWith)
25. Cancellation token
26. Всегда ли имеет смысл применять Task – потоковое голодание
27. Примитивы синхронизации – Monitor, Semaphore. Почему в lock нельзя использовать асинхронный код. Почему в lock нельзя использовать структуру. Разница системных и Slim версий.
28. Класс Interlocked
29. Потокобезопасные коллекции – зачем нужны, когда используются (ConcurrentDictionary, ConcurrentBag).

**Блок 2: ООП**

1. Инкапсуляция, наследование, полиморфизим
2. Virtual – переопределение
3. New – перекрытие
4. Ковариантность, контрвариантность
5. Абстрактный класс, abstract – гарантия реализации
6. Интерфейс
7. Абстрактный класс vs интерфейс – когда что использовать
8. Паттерны стратегия, шаблонный метод

Задачи:

Text

Description automatically generated

Что будет выведено и почему?

Ответ: From Base, From Derived. В .net приоритетным считается собственный метод, а не наследованный.

A screenshot of a computer

Description automatically generated

Ответ:

From A

From B

From B

From C

**Блок 3: Паттерны**

1. Фабричный метод – что и как используется
2. Декоратор
3. Какие знаешь возможные методы реализации publisher-subscriber? (События, System.Threading.Channel, шина данных на основе Mediator, RabbiqMq, Redis)
4. DI – зачем нужно
5. SOLID

**Блок 4: ASP.NET**

1. HTTP прокотол – как устроен, альтернативы
2. REST – зачем нужен, альтернативы
3. JSON,XML
4. Как устроен конвеер в ASP.NET
5. HttpContext
6. Встроенный DI контейнер – как работает

**Блок 5: Базы данных**

1. Реляционные БД. 3НФ.
2. Порядок выполнения запросов (FROM, WHERE, GROUP BY, HAVING, SELECT)
3. Как работает агрегация в SQL. Group By.
4. Назови все способы соединения таблиц (join, left join, right join, full join, cross join, cross apply, outer apply)
5. Оконные функции over() by