

10 взаимосвязанных задач в стиле RPG “Мастер кода”, где пошагово создается система симуляции гильдии магов, изучая **async/await** и **многопоточность**.

Проект: “Гильдия Магов Асинхронум”

Задача 1: Инициализация гильдии Создай класс Guild с методом InitializeGuildAsync, который асинхронно загружает из файла (эмулировать задержкой) список магов и их базовые характеристики. Метод должен возвращать Task<List>.

Задача 2: Энергия магов Реализуй систему восстановления маны. Для каждого мага в списке запусти фоновую задачу, которая каждые 3 секунды увеличивает его текущую ману на 10 единиц (но не больше максимума). Используй Task.Run и бесконечный цикл с await Task.Delay.

Задача 3: Мониторинг состояния Создай метод DisplayMagesStatusAsync, который раз в 5 секунд асинхронно выводит в консоль таблицу со статусом всех магов (имя, текущая мана, состояние). Метод должен работать независимо от основного потока.

Задача 4: Магический тренинг Реализуй метод StartTrainingAsync, принимающий имя мага и количество повторений. Метод должен асинхронно выполнять серию “тренировочных заклинаний” (эмулировать Task.Delay), выводя информацию о каждом выполненном заклинании.

Задача 5: Соревнование магов Создай метод StartDuelsAsync, который запускает параллельные “дуэли” между парами магов. Каждая дуэль - это асинхронная операция, где маг тратит ману и через случайное время получает результат (победа/поражение). Используй Task.WhenAll для ожидания всех дуэлей.

Задача 6: Магический тотем Реализуй класс MagicTotem с методом ActivateTotemAsync, который периодически (каждые 10 секунд) асинхронно усиливает всех магов, временно увеличивая их максимальную ману. Тотем должен работать как независимая фоновая задача.

Задача 7: Поиск артефактов Создай метод SearchArtifactsAsync, который запускает 5 параллельных “поисковых миссий”. Каждая миссия должна независимо и асинхронно искать артефакт, возвращая результат после случайной задержки. Собери результаты с помощью Task.WhenAll.

Задача 8: Ритуал усиления Реализуй метод PerformRitualAsync, в котором 3 мага одновременно начинают ритуал. Ритуал считается успешным, только если все маги завершили свои части заклинания в течение 5 секунд. Используй Task.WhenAll с таймаутом.

Задача 9: Магический совет Создай метод `CallCouncilMeetingAsync`, который должен последовательно (но асинхронно) опросить 4 магов-советников. Каждый советник “обдумывает” ответ случайное время. Используй `await` в цикле для последовательного выполнения.

Задача 10*: Битва с боссом Реализуй финальную битву - метод `StartBossFightAsync`. Все маги атакуют босса параллельно, каждый в своем темпе. Битва заканчивается когда здоровье босса опустится до нуля. Используй непрерывный мониторинг состояния босса с параллельными атаками.

Этот набор задач постепенно усложняется и охватывает *ключевые* аспекты *асинхронного* программирования, от простых `Task.Delay` до сложных сценариев с параллельным выполнением и координацией задач.

Задачи со звездочкой (*) имеют некоторую проблему. Как думаете какую?