

Выполните все приведенные ниже задания (в этом модуле нет complex task, все задания обязательны):

Для этого вам понадобится загрузить шаблон [AsyncAwait.Tasks](#).

1. Asynchronous calculation and cancellation tokens

(AsyncAwait.Task1.CancellationTokens.csproj)

Перед выполнением необходимо изучить:

- [Module 2. Asynchronous programming](#)

Задание

Изучите код данного приложения для расчета суммы целых чисел от 0 до N, а затем измените код приложения так, чтобы выполнялись все нижеследующие требования:

- Расчет должен производиться асинхронно.
- N задается пользователем из консоли. Пользователь вправе внести новую границу в процессе вычислений, что должно привести к перезапуску расчета.
- При перезапуске расчета приложение должно продолжить работу без каких-либо сбоев.

2. ASP MVC challenge

(AsyncAwait.Task2.CodeReviewChallenge.csproj, AsyncAwait.Task2.CodeReviewChallenge.Tests.csproj)

Перед выполнением необходимо изучить:

- [Asynchronous programming. Part III Dos and Don'ts.mp4](#)

Задание

Допустим, вы эксперт в асинхронном программировании. Представьте, что вас попросили сделать code review ASP.Net Core приложения на предмет корректной реализации асинхронных операций.

Что делает приложение

Сам сайт содержит три страницы, на которые можно перейти, кликая по пунктам главного меню: Home, Privacy, Help. Помимо этого, на каждой странице считается статистика посещений:

Welcome!

This application has the following working functions:

- Menu: Home
- Menu: Privacy
- Menu: Help
- A counter of visits

This page has been visited **3** time(-s) since the launch of the service.

2019 - AsyncAwait.CodeReviewChallenge - [Privacy](#)

Вероятно, код подсчета статистики у разработчиков работал неправильно, и поэтому они применили несколько workaround-ов, что привело к очень долгой загрузке страниц.

Что нужно сделать:

1) Проанализируйте код приложения **AsyncAwait.CodeReviewChallenge**, выпишите все замечания по асинхронному коду, дайте рекомендации по их устранению.

2) Исправьте код в соответствии с собственными рекомендациями, убедитесь, что после исправлений приложение работает стабильно (возможно, это удобнее делать в отдельной ветке, чтобы было удобно демонстрировать оба задания).

В решении также есть проект CloudServices. В ней симулировано обращение к какому-то сторонним сервисам. Это внешняя библиотека, согласно заданию, у вас нет возможности изменять ее код. Все изменения должны быть только в проекте AsyncAwait.CodeReviewChallenge.

Обсудите рекомендации и полученный результат с ментором. Будьте готовы дать пояснения по поводу того, что при этом происходит «под капотом».

Оценка по модулю должна определяться по двум заданиям и во многом должна зависеть от того, насколько качественно проведена экспертиза, и насколько корректно mentee понимает суть найденных замечаний и предложенных рекомендаций.