**НЕГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ ЧАСТНОЕ** **УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ** **«МОСКОВСКИЙ ФИНАНСОВО-ПРОМЫШЛЕННЫЙ УНИВЕРСИТЕТ** **“СИНЕРГИЯ”»**

|  |  |  |
| --- | --- | --- |
| **Факультет/Институт** |  | колледж |
|  |  | (наименование факультета/ Института) |
| **Направление/специальность** |  | ИС и программирование |
| **подготовки:** |  | (код и наименование направления /специальности подготовки) |
| **Форма обучения:** |  | очная |
|  |  | (очная, очно-заочная, заочная) |
|  |  |  |

**Отчет по лабораторной работе №7**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **на тему** |  | Нагрузочное тестирование, стрессовое тестирование | | |
|  |  | (наименование темы) | | |
|  |  |  | | |
| **по дисциплине** | | |  | Тестирование информационных систем |
|  | | |  | (наименование дисциплины) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Обучающийся** |  | Масевич Екатерина Александровна |  |  |
|  |  | (ФИО) |  | (подпись) |
| **Группа** |  | ДКИП-311 |  |  |
|  |  |  |  |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Преподаватель** |  | Сибирев Иван Валерьевич |  |  |
|  |  | (ФИО) |  | (подпись) |

**Москва 2025 г.**

**Лабораторная работа №7. «**Нагрузочное тестирование, стрессовое тестирование**»**

**Цель работы:**

Цель работы: изучение нагрузочного тестирования ПО. Результатом работы является отчет, в котором должны быть приведены исходные коды программы, результаты нагрузочного тестирования ПО.

**Пример кода для нагрузочного тестирования**

using System;

using System.Net.Http;

using System.Threading.Tasks;

class LoadTest

{

private static readonly HttpClient client = new HttpClient();

static async Task Main(string[] args)

{

int numberOfRequests = 100; // Количество запросов

Task[] tasks = new Task[numberOfRequests];

for (int i = 0; i < numberOfRequests; i++)

{

tasks[i] = SendRequestAsync();

}

await Task.WhenAll(tasks);

Console.WriteLine("Все запросы завершены.");

}

private static async Task SendRequestAsync()

{

try

{

HttpResponseMessage response = await client.GetAsync("https://yourapi.com/endpoint");

response.EnsureSuccessStatusCode();

string responseBody = await response.Content.ReadAsStringAsync();

Console.WriteLine($"Ответ: {responseBody}");

}

catch (HttpRequestException e)

{

Console.WriteLine($"Ошибка запроса: {e.Message}");

}

}

}

**Подготовка тестового окружения**

1. **Настройка окружения**:
   * Убедитесь, что все необходимые сервисы и базы данных работают и доступны.
   * Получите доступ к API или веб-приложению, которое вы собираетесь тестировать.
2. **Создание тестовых случаев**:
   * Определите ключевые сценарии использования, которые требуют взаимодействия между модулями.

**Выполнение тестов**

* Запустите программу, чтобы инициировать нагрузочное тестирование. В этом примере программа отправляет 100 параллельных запросов к указанному API.

**Анализ результатов**

* После выполнения теста проанализируйте результаты, чтобы выявить возможные узкие места или ошибки в производительности.

**Документирование**

* Запишите все найденные ошибки и результаты тестирования для дальнейшего анализа и улучшения приложения.

Этот подход поможет эффективно провести нагрузочное тестирование вашего ПО с помощью C# в Visual Studio 2022.