# Звіт до лабораторної 2

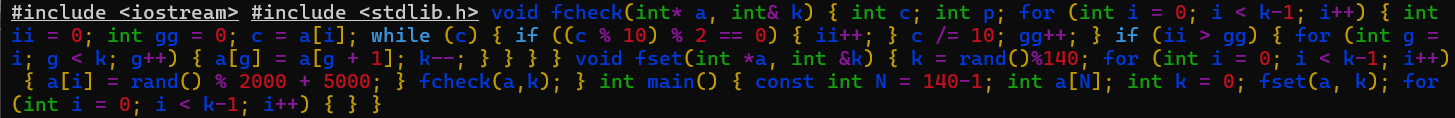
Студента групи ТТП-32

Остренка Олександра

## 1. Умова лабораторної: Изображение выглядит как текст, снимок экрана, Шрифт, документ Автоматически созданное описание

## 2. Опис виконання:

Лабораторна була виконана на мові С# за допомогою фреймворку xUnit.

* Суть програми, що тестується.  
   Програма реалізує примітивний лексичний аналізатор, який виділяє та розмальовує відповідні лексеми для коду, написаного мовою С.  
    
   Програма складається з трьох ключових етапів – зчитування фалу, підготовка тексту для аналізу (видалення всіх коментарів, переносів строки та зайвих пробілів) та сам лексичний аналіз.
* Тестові методи:  
  Для наочності автор розбив тести на два класи – xUnitTest1 та xUnitTest2. В першому зосереджені тести, що тестують допоміжний функціонал програми (читання файлу та видалення всього зайвого – коментарів, переносів строк і тд). В другому класі зберігаються тести, що тестують основну функцію, яка виділяє та розмальовує відповідні лексеми. Також, для наочності, в кожен метод добавлено Delay в одну секунду.

**xUnitTest1:**

* + Setup.  
    Присутній один setup-метод, що викликається перед кожним тестовим методом і ініціалізує лексер.  
    Изображение выглядит как текст, Шрифт, снимок экрана, дизайн

    Автоматически созданное описание
  + Параметризований тест, що перевіряє правильність роботи метода, що зчитує зміст файлу:  
    Изображение выглядит как текст, снимок экрана, Шрифт

    Автоматически созданное описание
  + Тести для методу, що перевіряє правильність видалення коментарів:  
    Изображение выглядит как текст, снимок экрана, Шрифт

    Автоматически созданное описание
  + Більш складний метод, що перевіряє правильність видалення коментарів:  
    Изображение выглядит как текст, снимок экрана, Шрифт

    Автоматически созданное описание

**xUnitTest2:**

* Setup + Dispose:  
  Изображение выглядит как текст, снимок экрана, Шрифт

  Автоматически созданное описание
* Метод, що тестує виключення для функції виділення та фарбування лексем:  
  Изображение выглядит как текст, Шрифт, линия, снимок экрана

  Автоматически созданное описание
* Параметризований метод, що тестує правильність роботи функції, що виділяє та розфарбовує лексеми:  
  Изображение выглядит как текст, снимок экрана, Шрифт

  Автоматически созданное описание

## 3. Конфігурації

* Конфігурація, що має на меті запустити всі тести паралельно:  
  Изображение выглядит как текст, снимок экрана, Шрифт

  Автоматически созданное описание  
  Для перевірки, введемо в термінал команду -  
  «dotnet test --settings:Config1.test.runsettings».  
  
* Щоб було з чим зрівняти попередній результат – конфігурація, що запускає тести послідовно:  
  Изображение выглядит как текст, снимок экрана, Шрифт

  Автоматически созданное описание   
  Для перевірки, введемо в термінал команду -  
  «dotnet test --settings:Config2.test.runsettings».  
    
  Бачимо, що послідовно вони виконуються на дві секунди довше – це саме ті два тесту з другого класу, що виконувались не синхронно з першим класом, а лише після нього.
* Конфігурація, що запускає тільки якусь групу тестів (в даному випадку було виділено тільки дві групи – «Preparation» та «MainPart»).  
  Изображение выглядит как текст, Шрифт, снимок экрана

  Автоматически созданное описание   
  Для перевірки, введемо в термінал команду -  
  «dotnet test --settings:Config3.test.runsettings».  
    
  І дійсно, викликаються тільки шість тестів з першого класу – методи так званої «підготовки».