# Лабораторна робота №1.

**Мета:**

* Розробка власних утилітарних класів.
* Набуття навичок вирішення прикладних задач з використанням масивів і рядків.
* Реалізація діалогового режиму роботи з користувачем в консольних програмах мовою Java.

# Вимоги

1. Розробити та продемонструвати консольну програму мовою Java в середовищі Eclipse для вирішення прикладної задачі за номером, що відповідає збільшеному на одиницю залишку від ділення на 15 зменшеного на одиницю номера студента в журналі групи.
2. Використовуючи програму рішення завдання відповідно до прикладної задачі забезпечити обробку команд користувача у вигляді текстового меню:
   1. введення даних;
   2. перегляд даних;
   3. виконання обчислень;
   4. відображення результату;
   5. завершення програми і т.д.
3. Забезпечити обробку параметрів командного рядка для визначення режиму роботи програми:
   1. параметр “-h” чи “-help”: відображається інформація про автора програми, призначення (індивідуальне завдання), детальний опис режимів роботи (пунктів меню та параметрів командного рядка);
   2. параметр “-d” чи “-debug”: в процесі роботи програми відображаються додаткові дані, що полегшують налагодження та перевірку працездатності програми: діагностичні повідомлення, проміжні значення змінних, значення тимчасових змінних та ін.
4. При вирішенні прикладних задач використовувати латинку .
5. Продемонструвати використання об’єктів класу StringBuilder або StringBuffer .
6. Застосувати функціональну (процедурну) декомпозицію - розробити власні утилітарні класи (особливий випадок допоміжного класу, див. Helper Class ) та для обробки даних використовувати відповідні статичні методи.
7. Забороняється використовувати засоби обробки регулярних виразів: класи пакету java.util.regex ( Pattern , Matcher та ін.), а також відповідні методи класу String ( matches , replace , replaceFirst , replaceAll , split ).

# Розробник

Зінько Павло, КН-108, номер варіанту – 12.

# Задача

Ввести текст. Текст розбити на речення. Для кожного речення вивести слова, у яких перша та остання літери співпадають. Результат вивести у вигляді таблиці.

# Опис програми

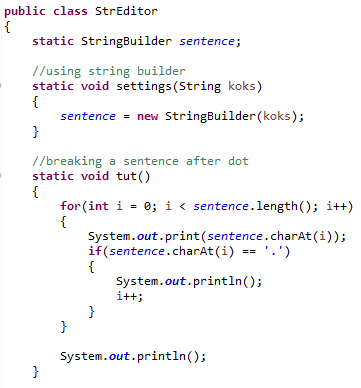
Програма розбиває текст на речення, перевіряє кожне слово та обробляє його відповідно до завдання, добавлено меню, яке викликається за допомогою консольної команди -h(режим допомоги та інформації) та -d(debug) яке виведе на екран всі неправильні слова.

# Засоби ООП

Для виконання завдання був використаний утилітарний клас StrEditor, який містив певні статичні поля і методи необхідні для перевірок.

# Ієрархія та структура класів

1. Клас Point, який містить 1 функцію – main, тобто саме меню.
2. Клас StrEditor з статистичними полями та декількома функціями: (settings, tut та WordChecker, вигляд якого показано в наступному пункті)



# Важливі фрагменти програми.

Функція перевірки слів:


# Варіанти використання

Дана програма може використовуватись для перевірки слів у тексті

# ВИСНОВКИ

У ході роботи ми розробили власний утилітарний клас, набули навичок використання StringBuilder, розробили діалоговий режим роботи з користувачем.