# Лабораторна робота 3 «Налагодження програмного забезпечення»

## Мета роботи

Головною метою цієї лабораторної роботи є надання студентам практичних навичок у налагодженні програмного забезпечення з використанням різноманітних інструментів і технік. Студенти мають реалізувати специфічну частину функціоналу своєї програми, вивантажити здійснені зміни до репозиторію, а потім здійснити комплексні заходи щодо налагодження свого коду. Це включає в себе застосування таких інструментів налагодження, як точки зупинки і точки зупинки з умовами, покрокове виконання програми та розрахунок часу виконання, перегляд і зміна значень змінних у процесі роботи програми, а також зміну коду на льоту без необхідності повторного запуску програми.

Ця робота спрямована на розвиток здатності студентів ефективно використовувати сучасні інструменти для ідентифікації, локалізації та виправлення помилок у програмному коді. Ключовим аспектом роботи є засвоєння методик виявлення помилок, які зустрічаються під час реалізації програмного проекту, та формування умінь швидкого реагування на них, забезпечуючи високу надійність та якість кінцевого продукту.

## Опис завдання

У рамках цієї лабораторної роботи кожен студент займатиметься реалізацією визначеної частини функціоналу свого програмного проекту. Це може включати, але не обмежується, розробкою нових функцій, оптимізацією існуючого коду або інтеграцією з зовнішніми сервісами або бібліотеками. Основна увага має бути зосереджена на дотриманні найкращих практик програмування та ефективному використанні інструментів налагодження для забезпечення високої якості програмного коду.

Після завершення реалізації, студенти повинні вивантажити здійснені зміни до свого репозиторію на GitHub. Наступний етап роботи включає налагодження програмного коду. Студентам необхідно використовувати наступні інструменти налагодження, які є в інтегрованому середовищі розробки Visual Studio:

* **Точки зупинки:** дозволяють призупинити виконання програми в певному місці.
* **Точки зупинки з умовами:** умовні точки зупинки, які активуються лише за певних умов.
* **Покрокове виконання:** дозволяє виконувати програму крок за кроком (як в прямому, так і в зворотному напрямках) для детального аналізу.
* **Зміна значень змінних:** можливість змінювати значення змінних під час налагодження для тестування різних сценаріїв.
* **Зміна коду на льоту:** внесення змін до коду без необхідності повторного запуску програми, що забезпечує швидку ітерацію та тестування.
* **Перегляд значень змінних:** використання вікон налагодження для перегляду та аналізу значень змінних у реальному часі, що сприяє глибшому розумінню стану програми.
* **Розрахунок часу виконання:** вимірювання та аналіз часу виконання окремих сегментів коду або програми в цілому для ідентифікації та оптимізації потенційних вузьких місць у продуктивності.

Метою застосування цих інструментів є ідентифікація та виправлення помилок, оптимізація логіки програми та забезпечення її надійної та ефективної роботи. Результати налагодження, включаючи опис використаних методик, виявлених помилок, способів їх виявлення та методів вирішення, а також застосовані оптимізації, слід детально відобразити у звіті про виконану роботу.

## Звіт про виконану роботу

Звіт про виконану роботу повинен починатися з вступу, де коротко описується мета лабораторної роботи та зазначається загальна інформація про частину програми, яка була реалізована. У цьому розділі важливо вказати основні цілі та очікувані результати роботи над проектом.

Далі слід детально описати процес реалізації визначеної частини програмного забезпечення. Цей опис повинен включати інформацію про технічні деталі реалізації, такі як використані алгоритми, структури даних та основні методики програмування. Опис реалізації має надати чітке уявлення про те, як були вирішені задачі та які технічні рішення були обрані.

У наступній частині звіту необхідно розглянути використані інструменти налагодження. Опис повинен охоплювати різноманітні інструменти налагодження, застосовані під час роботи, включаючи, але не обмежуючись, точками зупинки, покроковим виконанням, переглядом та зміною значень змінних, а також розрахунком часу виконання програми. Важливо детально описати, як кожен з цих інструментів був використаний для виявлення та виправлення помилок, оптимізації логіки програми та забезпечення її ефективної роботи.

Завершальна частина звіту має містити аналіз виконаної роботи, включаючи огляд виявлених та виправлених помилок, оцінку ефективності використаних методик. Також слід зазначити, яким чином виконана робота вплине на подальший розвиток проекту, та висловити міркування щодо можливих напрямків оптимізації та вдосконалення програми.