Міністерство освіти і науки України Національний технічний університет України «Київський політехнічний інститут імені Ігоря Сікорського» Факультет інформатики та обчислювальної техніки Кафедра обчислювальної техніки

Курсова робота

з дисципліни

«Компоненти програмної інженерії. Частина 4. Якість та тестування програмного забезпечення»

Виконав:

студент групи ІП-05

Гапій Денис Едуардович

номер залікової : 0504

Перевірив:

Таран В. І.

Київ 2022

Тема: «Якість та тестування програмного забезпечення»

Мета: Провести рефлексію та підбити підсумки вивченого матеріалу за навчальний рік.

Лабораторна робота №1:

Суть роботи: Навчитися створювати Unit-тести. Продемонструвати базові навички роботи з системою контролю версій Git.

Необхідно було реалізувати консольний застосунок, що є аналогом всесвітньо відомої гри «Гра́ життя́» -- клітинний автомат, винайдений англійським математиком Джоном Конвеєм 1970 року, що базується на генетичних законах Конвея. Складається з основниз 4-ьох правил:

- якщо в живої клітини два чи три живих сусіди – то вона лишається жити;
- якщо в живої клітини один чи немає живих сусідів – то вона помирає від «самотності»;
- якщо в живої клітини чотири та більше живих сусідів – вона помирає від «перенаселення»;
- якщо в мертвої клітини рівно три живих сусіди – то вона оживає.

Важливим було опанувати усі види тестування програмного забезпечення, від Модульних й до Інтеграційних чи End-To-End тестів.

Дізнався / Опанував	● Генетичні закони Конвея ● GitHub Actions ● .NetCore збірка проєктів та взаємодія через CL ● MarkDown формати ● Розробка алгоритмів побудованих на взаємодії з матрицями ● Опрацювання помилок в мові C# ● Unit-tests ● Фреймворк XUnit
Були труднощі з ...	● Створенням логіки станів, та пошуку сусідів
Рівень освоєння матеріалу	10/10

👉РЕПОЗИТОРІЙ👈

Лабораторна робота №2:

Суть роботи: Покращення навичок зі створення Unit-тестів. Дізнатися про базові положення ‘Екстремального’ програмування.

Алгоритмічне завдання полягало у реалізовані спрощеного клавішного калькулятора.

Клавіші калькулятора:

"0" ... "9" — введення цифри

"+", "-", "*", "/" — виконання арифметичної операції над цілими числами

"=" — виконання операції

Для спрощення виконання завдання вводимо наступні обмеження, які відсутні у звичайному калькуляторі:

- обчислення виконуються над цілими числами (int, не float, не double),
- користувач може тільки ввести послідовність цифр, потім операцію, потім знову цифри, потім "=". Будь-яка інша послідовність клавіш є забороненою.

Також потрібно було попрацювати над форматом вводу та виводу

Дізнався / Опанував	● Test Driven Development ● Simple design based at abstractions ● Архітектурний підхід до програмного коду ● Більшу к-сть правил створення модульних тестів
Були труднощі з ...	● Парним програмуванням, адже для цього потрібно мати вільну людину ● Рефакторингом, адже для цього потрібна натхнення
Рівень освоєння матеріалу	10/10

👉Р Е П О З И Т О Р І Й👈

Лабораторна робота №3:

Суть роботи: Навчитися створювати Mock-и / Fake-и.

Необхідно було реалізувати консольний застосунок, що є аналогом всесвітньо відомої гри «Тетріс». Потрібно додатково чітко розділити програму на шари:

- “логічний” - тут реалізована логіка гри де ви оперуєте з абстракціями, які ви ж і створили (ігрове поле, код пов’язаний з правилами гри, тощо)
- “введення/виведення” - тут реалізується комунікація вашої програми з зовнішнім середовищем, де ви працюєте з наданими абстракціями (стандартний виводом, файловою системою)
- “комунікаційний” - шар який забезпечує комунікацію між “логікою” і “введенням/виведенням”. У цьому шарі повинен бути реалізований шаблон “DI (dependency injection)” який дозволяє підміняти шар введення/виведення у тестовому оточенні.

Дізнався / Опанував	● Принципи Code Review ● Покриття тестами та відсоткова статистика ● Створювати Fake ● DI (dependency injection) ● Goodhart's law
Були труднощі з ...	● Правильною реалізацією класів для `Впровадження залежностей`, попри те що розумію, як працюють класи, але все ж саме цей ‘шаблон’ для мене досі залишається складним (у запроваджені, а не у розумінні концепції)
Рівень освоєння матеріалу	6/10

👉Р Е П О З И Т О Р І Й👈

Лабораторна робота №4:

Суть роботи: Навчитися створювати Mock-и за допомогою спеціальних бібліотек.

Необхідно було доробити та оптимізувати консольний застосунок, що є аналогом всесвітньо відомої гри «Тетріс» зроблену у лабораторній #3 і додати параметр при передачі якого, буде виведено не тільки останній екран гри, а і екрани на кожному кроці гри. При цьому потрібно зберегти і початковий режим роботи (коли виводиться тільки останній екран).

Дізнався / Опанував	● Автоматично згенеровані Моки (замінники класів/інтерфейсів) ● Fakes | Stubs ● Бібліотека Moq ● Command line args / flag parsing
Були труднощі з ...	● Жодних, було навіть цікаво дізнатися та попрактикуватися з новою бібліотекою, використовуючи сторонню документацію
Рівень освоєння матеріалу	7/10

👉Р Е П О З И Т О Р І Й👈

Висновок:

Під часа опанування дицспиліни ‘Компоненти програмної інженерії. Частина 4. Якість та тестування програмного забезпечення’ я успішно здобув велику кількість нових навичок та покращив старі у програмуванні, поетапної розробки застосунку, взаємодії в команді, тестування існуючого коду та створення тестів для якісного написання подальшого коду, все це підкріплено фундаментальними знаннями про Unit / Integration / End To End тести.

Знання про: екстремального програмування, ревью програмного коду, парне програмування, покриття тестів, модульність, абстракції, системи введення / виведення, впровадження залежностей, моки / стаби / фейки, та багато інших принципів та практик, допомогли мені сформуватися як фахівця, що може стати експертним у тестуванні програмного забезпечення, взаємодії з іншими розробниками, використанні сторонніх, але допоміжних бібліотек / фреймворків.

Витраченого часу на цю дисципліну - дійсно не шкода.