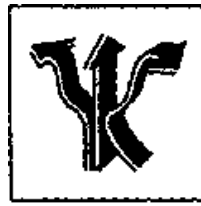


**ПРИВАТНЕ АКЦІОНЕРНЕ ТОВАРИСТВО «ВИШИЙ НАВЧАЛЬНИЙ ЗАКЛАД  
«МІЖРЕГІОНАЛЬНА АКАДЕМІЯ УПРАВЛІННЯ ПЕРСОНАЛОМ»**



**МАУП**

**НАВЧАЛЬНА ПРОГРАМА**  
дисципліни  
**" МЕТОДИ ТА ЗАСОБИ ТЕСТУВАННЯ ПРОГРАМНОГО  
ЗАБЕЗПЕЧЕННЯ "**  
(для магістрів)

Київ 2018

Підготовлено проф. кафедри комп'ютерних інформаційних систем та технологій  
М.П.Дяченком

Затверджено на засіданні кафедри комп'ютерних інформаційних систем та технологій  
(Протокол № 1 від 22.08.2018)

Схвалено Вченою радою Міжрегіональної Академії управління персоналом

**Дяченко М.П.**, Навчальна програма дисципліни “ Методи та засоби тестування програмного забезпечення.” (для освітньо-кваліфікаційного рівня ‘магістр’). — К.: МАУП, 2018. — 20 с.

Навчальна програма містить пояснювальну записку, тематичний план дисципліни “ Методи та засоби тестування програмного забезпечення ”, теми контрольних, лабораторних робіт, вказівки до самостійного вивчення програмного матеріалу та виконання контрольних і лабораторних робіт, питання для самоконтролю та список літератури.

© Міжрегіональна Академія управління персоналом (МАУП), 2018

## **ПОЯСНЮВАЛЬНА ЗАПИСКА.**

**Навчальний курс** є складовою частиною загальної програми підготовки спеціалістів в області інформаційних технологій. Курс підпорядкований вивченню засобів контролю та забезпечення якості програмних виробів через наскрізне тестування всього процесу розробки систем в цілому і їх окремих компонент. Увага курсу зосереджена на фундаментальних положеннях дисципліни: проблемах надійності та якості програмних виробів, а також принципах, критеріях, методах і засобах тестування як інструменту розв'язання проблем якості програмного забезпечення.

**Метою курсу** є формування у студента правильного розуміння завдання тестування у його звязку з надійністю та якістю програмного продукту, а також систематизація знань стосовно способів досягнення сприйняттого рівня надійності та якості продукту.

**Завдання курсу** полягає у розкритті змісту тестування як окремого зрізу процесу розробки програмного продукту, впливу тестування на показники якості продукту, а також у формуванні навичок з організації і проведення тестування програмного забезпечення довільного типу і рівня складності.

Навчальний матеріал дисципліни структурований за модульним принципом і складається з чотирьох навчальних модулів:

- **навчальний модуль №1. «Програмний продукт (ПП) та його властивості »,**
- **навчальний модуль №2. «Контроль якості ПП через тестування»,**
- **навчальний модуль №3. «Автоматизоване тестування»**
- **навчальний модуль №4. «Забезпечення якості ПП»**

кожен з яких є логічно завершеною, відносно самостійною, цілісною частиною навчальної дисципліни, засвоєння якої передбачає проведення модульної контрольної роботи та аналіз результатів її виконання.

### **У результаті вивчення дисципліни:**

- *студент повинен знати* основні принципи, критерії, стратегію та етапи, тестування для досягнення необхідних характеристик якості програмного продукту, ознайомитись із формалізмом в організації тестування та засобами автоматизації тестування;
- *студент повинен вміти* створювати набори тестів для програм як на базі текстів програм та їх специфікацій, оцінювати якість тестового набору, проводити тестування програм, аналізувати результати тестування та оформляти відповідну документацію.

Загальний обсяг дисципліни – 150 годин, 5 кредитів ЄКТС.

З них: 48 год лекції, 22 год практичні та семінарські заняття, 80 год самостійна робота.

Дана дисципліна є обов'язковою для вивчення.

**ТЕМАТИЧНИЙ ПЛАН ДИСЦИПЛІНИ  
« МЕТОДИ ТА ЗАСОБИ ТЕСТУВАННЯ ПРОГРАМНОГО  
ЗАБЕЗПЕЧЕННЯ»**

№/п.п.	Назва змістового модуля та теми
<b>Змістовий модуль I. Програмний продукт (ПП) та його властивості .</b>	
Тема 1	Програмний продукт та технології його створення.
Тема 2	Поняття і характеристики зовнішньої якості ПП
Тема 3	Внутрішня якість та супроводжуваність ПП.
<b>Змістовий модуль II Контроль якості ПП через тестування</b>	
Тема 4	Види тестування і проведення тестування.
Тема 5	Тестові сценарії.
Тема 6	Особливості тестування мобільних пристроїв та WEB застосунків
<b>Змістовий модуль III. Автоматизоване тестування.</b>	
Тема 7	Підходи та організація автоматизованого тестування
Тема 8	WEB-тестування та тестування програм мобільних пристроїв.
Тема 9	Модульне тестування і фреймворки модульного тестування
<b>Змістовий модуль IV. Забезпечення якості ПП</b>	
Тема 10	Аудит ІТ-команди
Тема 11	Тестування вимог
Тема 12	Тестування процесу розробки ПП в цілому
	<b>Разом годин : 150</b>

**ЗМІСТ ДИСЦИПЛІНИ**  
**« МЕТОДИ ТА ЗАСОБИ ТЕСТУВАННЯ ПРОГРАМНОГО**  
**ЗАБЕЗПЕЧЕННЯ»**

**Змістовий модуль I. Програмний продукт (ПП) та його властивості .**

**Тема 1. Програмний продукт та технології його створення.**

Програма як товар. Класифікація ПП. Вимоги до ПП. Життєвий цикл, етапи і моделі розробки програм з заданими властивостями. «Важкі» та гнучкі моделі. Характеристика моделей. Зв'язок проектування з тестуванням.

**Тема 2. Поняття та характеристики якості ПП**

Показники складності, коректності, стійкості, відновлюваності, мобільності, ефективності та надійності як складових якості, методи їх оцінки. Помилки, причини і їх джерела, стійкість до помилок. Метрики очікуваної кількості помилок. Дефекти та їх класифікація.

**Тема 3. Внутрішня якість ПП та супроводжуваність.**

Моделі, метрики і стандарти якості. Якість документації до ПП. Модульність та стиль оформлення тексту. Структурованість, рекомендації до іменування змінних і коментарів. Угорська нотація.

*Література [1-8]*

**Змістовий модуль II Контроль якості ПП через тестування.**

**Тема 4. Види тестування і організація тестування.**

Тестування програм, цілі і задачі. Види та рівні тестування. Методи тестування. Артефакти тестування. Критерії початку і кінця тестування. Оракули тестування. Стандарти контролю якості ПП.

**Тема 5. Тестові сценарії**

Структура тестового сценарію. Моделі (операційні, логіко-алгебраїчні та проміжні) і методи розробки тестових сценаріїв. Граничні точки і класи еквівалентності. Порядок розробки тестів. Класифікація дефектів і процедура тестування. Життєвий цикл дефектів.

*Література [5-8]*

## **Тема 6. Особливості тестування мобільних пристроїв та WEB застосунків**

Врахування розмірів екрана і його орієнтації, різноманітності ресурсів, touch - інтерфейсу та підтримка жестів, Особливості тестування WEB-сервісів. Тестування клієнт-серверної взаємодії та взаємодії з базою даних. Крос-браузерність і різноманітність клієнтів. Тестування WEB-форм на боці клієнта

### ***Змістовий модуль III. Автоматизоване тестування***

#### **Тема 7. Підходи і організація автоматного тестування**

Цілі і задачі автоматизованого тестування. Вибір методології і розробка тестів: модуль-базованих, керованих даними, ключовими словами, об'єктами, моделями. Вибір інструменту і формування середовища для тестування, розробка і запуск тестів.

#### **Тема 8. WEB-тестування і тестування програм мобільних пристроїв**

Засоби автоматизованого тестування (Silk, TestComplet, Selrniium, тощо). Selenium: склад, призначення і методика використання. Тестування в різних браузерах. Робота з мобільними пристроями. Використання сторонніх сервісів.

#### **Тема 9. Модульне тестування**

Цілі і задачі модульного тестування. Організація модульного тестування. Покриття тестами коду. Фреймворки doctest, unittest, pytest, можливості і методика використання. Особливості тестування знизу вверх та зверху вниз

*Література [9-18]*

### **Змістовий модуль IV. Забезпечення якості ПП**

#### **Тема 10. Аудит IT-команди**

Кваліфікація і перепідготовка кадрів. Розподіл обов'язків і стандартизація робочих місць. Стандартизація процесів. Організація взаємодії робочих місць. Використання case-засобів у процесі проектування та взаємодії.

Мережеві засоби підтримки взаємодії. Застосування agile – технологій. Scrum і kanban - технології.

### **Тема 11. *Тестування вимог***

Робота з вимогами. Зрілість і якість вимог. Організація, структурування, моделювання, трасування і інтеграція вимог. Контрольний список вимог. Тестування вимог до інтерфейсу і юзабіліті. Робота з вимогами в моделях agile. Покриття тестами вимог. Види покриттів.

### **Тема 12. *Тестування процесу розробки ПП.***

Тестування календарного плану розробки. Верифікація процесу розробки. Валідація компонент і програми в цілому. Неперервне тестування. Тестування зверху вниз та знизу вверх. Використання заглушок і драйверів.

*Література [18-22]*



## ТЕМАТИКА КОНТРОЛЬНИХ РОБІТ.

1. Підготовки до тестування вибраної програмної системи: опис системи, складання плану тестування.
2. Тестування за столом: аналіз специфікацій системи, структури системи, структур даних і потоків даних.
3. Аналіз методів тестування і вибір методів, адекватних завданням тестування вибраної системи.
4. Статичний і динамічний аналіз коду.
5. Тестування складових частин системи: визначення тест-кейсів, сценаріїв прогонки, здійснення прогонки, аналіз результатів.
6. Інтеграційне тестування системи : контроль правильності обміну повідомленнями між об'єктами, модулями, компонентами, що пройшли автономне тестування.
7. Інваріанти програм і методи доведення правильності програм.
8. Верифікація і валідація програм.
9. Тестування користувацького інтерфейсу
10. Ручне тестування: розробка моделі взаємодії системи з системним оточенням та тестових сценаріїв, розробка тестів, здійснення прогонки, аналіз результатів, реєстрація в журналі.
11. Опис і аналіз ручного тестування.
12. Підготовка і заповнення стандартних форм документів за результатами тестування.
13. Оцінка результатів тестування, оракули тестування.
14. Автоматизація тестування за допомогою скриптів : визначення ланцюжків тестування та структур даних в ланцюжках на основі аналізу ручного тестування, опис тестових послідовностей , обміну даними між окремими тестами засобами скриптової мови.
15. Автоматична генерація тестів: використання формальних мов для опису структури і поведінки системи (UML, SDL, MSC).

16. MSC-діаграми і MSC-тести. Автоматична генерація тестів на базі формального опису моделі системи і її поведінки у вигляді послідовності подій.
17. Selenium, ознайомлення з принципами роботи та системними можливостями.
18. Використання шаблонів (паттернів) для автоматизації тестування.
19. Автоматизація модульного тестування.
20. Тестування інсталляції.

### **ТЕМИ ЛАБОРАТОРНИХ РОБІТ.**

1. Розробка специфікацій на програму.
2. Тестування вимог.
3. Складання тест-плану.
4. Формування чек-ліст.
5. Три фази тестування.
6. Вимоги до критеріїв тестування.
7. Мутаційний критеріїв тестування.
8. Тестування за стратегією чорної скриньки.
9. Тестування за стратегією білої скриньки.
10. Тестування документації.
11. Оцінки покриття програми і проекту.
12. Покриття вимог.
13. Модульне тестування.
14. Інтеграційне тестування.
15. Системне тестування.
16. Розробка і опис тестового сценарію.
17. Автоматизація тестування засобами Selenium.
18. Розробка сценаріїв для TestCompleat.

## ВКАЗІВКИ ДО ВИКОНАННЯ КОНТРОЛЬНИХ РОБІТ

Задачами контрольної роботи є:

- систематизація і закріплення теоретичних та практичних фахових знань, виявлення уміння студента застосовувати ці знання при вирішенні конкретних наукових, технічних, економічних і виробничих задач;
- перевірка уміння студента самостійно освоювати та використовувати сучасні інформаційні технології, програмно-апаратні засоби обчислювальної техніки;
- розвинення у студента навичок ведення самостійного науково-практичного пошуку, оволодіння методикою дослідження й експериментування при вирішенні проблем і питань, поставлених на курсове проектування;
- закріплення знань і навичок виконання графічних робіт та інших конструкторських документів у відповідності до вимог і правил, встановлених державними стандартами, Єдиною системою конструкторської документації (ЄСКД), Єдиною системою проектної документації (ЄСПД), іншими чинними нормативно-технічними документами;

Головна мета контрольної роботи – визначення рівня теоретичної та практичної підготовки студента з курсу „ Якість програмного забезпечення та тестування ”.

Контрольна робота сприяє розвитку у студента творчої ініціативи і самостійності в проведенні аналізу, добору й обґрунтування найбільш раціональних інженерних рішень.

Контрольна робота надає студентіві таких навичок виконання виробничих завдань, які допоможуть йому швидко адаптуватися до умов праці у професійному колективі.

## СТРУКТУРА КОНТРОЛЬНОЇ РОБОТИ

**Структура** контрольної роботи складається за наступних розділів:

Розділ 1. Відповіді на теоретичні питання (15 питань)

Розділ 2. Розв'язання практичного завдання за індивідуальним варіантом.

Список використаних літературних джерел

Додатки

Для **успішного виконання практичного завдання** необхідно:

1. Побудувати автотест за індивідуальним варіантом (варіант співпадає з номером студента за переліком в журналі) що складається з 3-6 авто тестів.
2. Розробити перевірку правильності виконання тестів.

Результати виконання індивідуального практичного завдання додаються до контрольної роботи на електронному носії.

### КРИТЕРІЇ ОЦІНЮВАННЯ КОНТРОЛЬНОЇ РОБОТИ

Контрольна робота оцінюється у 100 балів, з них:

Вид роботи	Загальна кількість балів	Критерії оцінювання
Теоретичні питання (15)	0-60 балів	
Одне теоретичне питання	4 бали	4 бали – нараховується якщо вибрано варіант відповіді що найточніше відповідає питанню теста. відповідь на питання є повною, ґрунтовною та не містить зайвих відомостей із суміжних областей. 2 бали якщо, вибір відповіді має відношення до суті питання
Практичне завдання	40 балів	
Концептуальне проектування авто тестів	0-20 балів	15-20 балів нараховується коли концептуальна модель побудована у відповідності до вимог предметної області. 10-15 балів нараховується, коли є незначні помилки у 0-10 балів нараховується коли є істотні помилки при побудові системи автоматичного тестування.
Повнота тестування (достатня кількість тестів для перевірки функціональності)	0-15 балів	10-15 балів нараховується коли автотести знаходять всі основні помилки 5-10 балів нараховується, коли є незначні помилки у написанні системи 0-5 балів нараховується коли є істотні помилки у написанні системи автотестування.
Проектування перевірок	0-5	Загальна кількість перевірок має бути не менша за 2 на кожний тест..

## ПИТАННЯ ДО САМОКОНТРОЛЮ.

1. Яку програмну систему можна назвати програмним продуктом ?
2. Які етапи розробки програмного продукту ви знаєте та наведіть їх характеристику?
3. Що таке – життєвий цикл ПП?
4. Перерахуйте основні властивості ПП?
5. Що таке – сценарій використання ПП та параметри сценарію?
6. Дайте визначення понять якості та критеріїв якості програмної системи?
7. Що означають поняття забезпечення якості та контроль якості?
8. З чого розпочинати забезпечення якості програмної системи?
9. Перерахуйте рекомендації по забезпеченню якості.
- 10.Що таке – надійності системи? Дайте перелік та розкрийте зміст вимог до надійності.
- 11.В чому полягає різниця між надійністю та якістю ПП?
- 12.Які прояви характеризують роботу ненадійного ПП?
- 13.Визначіть поняття збою та відмови системи.
- 14.Що таке помилка, та які джерела помилок ви знаєте?
- 15.Наведіть класифікацію помилок.
- 16.Які методи виявлення помилок ви знаєте?
- 17.Що таке *заглушка* і яка її роль в тестуванні?
- 18.Перерахуйте показники надійності та розкрийте їх зміст.
- 19.Що таке – дефект системи, навіщо потрібно збирати інформацію про дефекти?
- 20.Дайте характеристику поняття ортогональної класифікації дефектів.
- 21.Дайте класифікацію ризиків програмного проекту.
- 22.Що таке модуль та компонента системи?
- 23.Чи можна побудувати надійну систему з ненадійних компонентів?  
Відповідь обґрунтуйте.
- 24.Розкрийте зміст поняття тестування.
- 25.Охарактеризуйте об'єкти та суб'єкт тестування.

26. Перерахуйте завдання і ролі тестування.
27. Викладіть необхідні і достатні умови проведення тестування.
28. Який зміст укладається в термін *тестування з поверненням*?
29. Опишіть процедуру регресійного тестування.
30. Перерахуйте артефакти тестування та розкрийте їх зміст.
31. Що входить до складу тестової документації?
32. Розкрийте зміст тест-плану.
33. Що таке *тестове покриття*?
34. Які методи і інструменти для визначення тестового покриття ви знаєте?
35. Що таке дефект, важливість дефекта? Перерахуйте градації дефекта.
36. Чи потрібно накопичувати дефекти? Відповідь обґрунтуйте.
37. Що таке пріоритет дефекта? Які градації має пріоритет дефекта?
38. Що таке баг-репорт?
39. Розкрийте зміст поняття життєвого циклу баг-репорта.
40. Яка різниця між тестом, тест-кейсом і тестовим покриттям?
41. Якими критеріями відбору тестів ви користуєтесь?
42. Що означає поняття тест-дизайн? Які техніки тест-дизайну ви знаєте?
43. Викладіть зміст поняття *стратегія тестування*.
44. Які тести входять до навантажувального тестування?
45. В чому ви вбачаєте різницю між забезпеченням якості, контролем якості та тестуванням?
46. Що таке цикл тестування?
47. Перерахуйте рівні тестування, наведіть їх характеристику.
48. Чи вбачаєте ви різницю між компонентним, модульним та процедурним тестуванням?
49. Розкрийте зміст системного тестування.
50. Яка різниця між системним та інтеграційним тестуванням?
51. Які види тестування ви знаєте?
52. Охарактеризуйте цілі і завдання функціонального тестування.
53. Які нефункціональні види тестування ви знаєте?

54. Перерахуйте техніки функціонального тестування.
55. Як вибирати серед різних технік тестування?
56. Яка різниця між навантажувальним та стресовим тестуванням?
57. Які цілі навантажувального тестування?
58. Викладіть зміст тестування надійності.
59. Коли слід закінчити тестування?
60. Співставте межі та можливості ручного і автоматизованого тестування.
61. Перерахуйте способи визначення повноти тестування
62. В чому різниця між тестуванням і відлагодженням?
63. Що означає *тестування за столом*?
64. Розкрийте зміст тестування на основі моделей.
65. Перерахуйте вимоги до методів тестування.
66. Поясніть зміст методу компонентного тестування
67. В яких випадках застосовується метод *білої скриньки*? В чому його суть?
68. Охарактеризуйте метод *чорної скриньки*.
69. Викладіть зміст тестування гілок.
70. Яке тестування називають мутаційними?
71. В чому різниця між санітарним та димовим тестуванням?
72. Викладіть суть регресійного тестування.
73. Дайте визначення автоматизованого тестування.
74. Перерахуйте стадії методики автоматизованого тестування.
75. Що є інструментом автоматизованого тестування? Які інструменти ви знаєте?
76. В яких випадках застосування автоматизованого тестування є доцільним (недоцільним)?
77. Які переваги у автоматизованого тестування?
78. Порівняйте повну і часткову автоматизацію.
79. Опишіть архітектуру набору автоматизованих тестів.
80. На яких стадіях життєвого циклу проводиться тестування?
81. Охарактеризуйте співвідношення між життєвим циклом і тестуванням.

82. Яка різниця між цілями та завданнями тестування?
83. Дайте визначення статичного та динамічного тестування.
84. Що таке відлагодження, верифікація та валідація?
85. Яка різниця між верифікацією, валідацією та тестуванням?
86. Які типи тестуючих систем ви знаєте?
87. Охарактеризуйте системи типу запис\відтворення, наведіть приклади таких систем.
88. Що представляють собою фреймворки doctest, unittest та pytest? Для чого вони призначені?
89. В чому суть використання типових рішень (паттернів) в автоматизації програмування?
90. Які засоби формалізації представлення даних в тестуванні ви знаєте?
91. Для чого може бути використана система StarUML? Які аналоги ви знаєте?



Відповідність підсумкової семестрової рейтингової оцінки в балах оцінці за національною шкалою та шкалою ECTS

Оцінка в балах	Оцінка за національною шкалою	Оцінка за шкалою ECTS	
		Оцінка	Пояснення
<b>90-100</b>	<b><i>Відмінно</i></b>	<b>A</b>	<b>Відмінно</b> (відмінне виконання лише з незначною кількістю помилок)
<b>82 – 89</b>	<b>Добре</b>	<b>B</b>	<b>Дуже добре</b> (вище середнього рівня з кількома помилками)
<b>75 – 81</b>		<b>C</b>	<b>Добре</b> (в загальному вірне виконання з певною кількістю суттєвих помилок)
<b>67 – 74</b>	<b>Задовільно</b>	<b>D</b>	<b>Задовільно</b> (непогано, але зі значною кількістю недоліків)
<b>60 – 66</b>		<b>E</b>	<b>Достатньо</b> (виконання задовольняє мінімальним критеріям)
<b>35 – 59</b>	<b>Незадовільно</b>	<b>FX</b>	<b>Незадовільно</b> (з можливістю повторного складання)
<b>1 – 34</b>		<b>F</b>	Незадовільно <b>(з обов'язковим повторним курсом)</b>

Підсумкова семестрова рейтингова оцінка в балах, за національною шкалою та шкалою ECTS заноситься до заліково-екзаменаційної відомості, навчальної картки та залікової книжки студента.

Підсумкова семестрова рейтингова оцінка заноситься до залікової книжки та навчальної картки студента, наприклад, так: **92/Відм./A**, **87/Добре/B**, **79/Добре/C**, **68/Задов./D**, **65/Задов./E** тощо..

Підсумкова рейтингова оцінка з дисципліни визначається як середньоарифметична оцінка з підсумкових семестрових рейтингових оцінок у балах (з цієї дисципліни – за перший та другий семестри) з наступним її переведенням в оцінки за національною шкалою та шкалою ECTS.

## СПИСОК ЛІТЕРАТУРИ.

1. Про затвердження загальних вимог до програмних продуктів.  
Постанова Кабінету Міністрів України від 12 серпня 2009 р. N 869
2. Процеси життєвого циклу програмного забезпечення. ДКА України  
СОУ-Н ДКА 0061:2012
3. Марголин Л. Н. Компьютерные методы обработки информации.  
Интернет издание. <http://hi-edu.ru/e-books/xbook709/01/title.htm>
4. Майерс Г. Надежность программного обеспечения. М.: Мир, 1980
5. Куликов С. С. Тестирование программного обеспечения. Базовый  
курс. Интернет-издание 2017
6. Канер С., и др. Тестирования программного обеспечения. К.,  
Диасофт, 2001.
7. Майерс Г. Искусство тестирования программ. М.: Финансы и  
статистика, 1982.
8. Савин Р. Тестирование Дот Ком. М.: «Дело», 2007.
9. Бахтизин В.В. Автоматизация тестирования программного  
обеспечения. Минск, БГУИР 2012
10. Особенности тестирования WEB-приложений. <http://quality-lab.ru/key-principles-of-web-testing/>
11. Мобильное тестирование: полное руководство.  
<http://www.cmsmagazine.ru/library/items/mobile/testing-mobile-apps/>
12. Тестирование по-пайтоновски. <https://dou.ua/lenta/articles/nose-tests-intro/>
13. Тестирование в Python [unittest]. <https://devpractice.ru/unit-testing-in-python-part-1/>
14. Python. Генерация юнит-тестов. <https://habrahabr.ru/post/192512/>
15. Python уроки: Введение в тестирование на Python.  
<https://pynsk.ru/blog/2016/02/10/intro-test/>

16. Автоматизация тестирования веб-приложения с использованием Selenium WebDriver, Python, и Behave/  
<https://habrahabr.ru/post/262929/>
17. `doctest` — Тестирование интерактивных примеров на Python.  
<http://python-lab.ru/documentation/27/stdlib/doctest.html>
18. Памятка по написанию тестов припомощи PyTest.  
<https://eax.me/pytest/>
19. Маглинец Ю.А. Анализ требований к автоматизированным информационным системам.  
<https://www.intuit.ru/studies/courses/2188/174/info>
20. Книберг Х. Скарин М. Scrum и Kanban: выжимаем максимум. Киев: InfoQ, 2011.
21. Мутилин В. С. Паттерны проектирования тестовых сценариев. Труды Института системного программирования РАН.  
<http://www.citforum.ru/SE/testing/>
22. Уиттакер Д. и др. Как тестируют в Гугл. Спб: Питерб 2014.

## ***ЗМІСТ***

<u>ПОЯСНЮВАЛЬНА ЗАПИСКА</u> .....	<b>Ошибка! Закладка не определена.</b>
<u>ТЕМАТИЧНИЙ ПЛАН</u> .....	<b>Ошибка! Закладка не определена.</b>
<u>ЗМІСТ ДИСЦИПЛІНИ</u> .....	<b>Ошибка! Закладка не определена.</b>
<u>ВКАЗІВКИ ЩОДО ВИКОНАННЯ КОНТРОЛЬНИХ РОБІТ</u> .....	11
<u>ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ</u>	<b>Ошибка! Закладка не определена.</b> 3
<u>СПИСОК ЛІТЕРАТУРИ</u> .....	<b>Ошибка! Закладка не определена.</b>