

Міністерство освіти і науки України  
Національний аерокосмічний університет ім. М. Є. Жуковського  
«Харківський авіаційний інститут»

Кафедра «Інженерії програмного забезпечення» (№ 603)

**ЗАТВЕРДЖУЮ**

Гарант освітньої програми  
 **І.Б. Туркін**  
(підпис) (ініціали та прізвище)

« 31 » 08 2021 р.

**СИЛАБУС ВИБІРКОВОЇ  
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

**Автоматизоване тестування програмного забезпечення**  
(назва навчальної дисципліни)

Галузь знань: 12 Інформаційні технології  
(шифр і найменування галузі знань)

Спеціальність: 121 Інженерія програмного забезпечення  
(код та найменування спеціальності)

Освітня програма: Інженерія програмного забезпечення  
(найменування освітньої програми)

**Форма навчання: денна**

**Рівень вищої освіти: перший (бакалаврський)**

**Силабус введено в дію з 01.09.2021 року**

**Харків – 2021 р.**

Розробник: Манжос Ю.С., доц. кафедри №603, канд. техн. наук, доц.  
(прізвище та ініціали, посада, науковий ступінь та вчене звання)



Силабус навчальної дисципліни розглянуто на засіданні кафедри інженерії програмного забезпечення (№ 603)

Протокол № 2 від « 31 » серпня 2021 р.

Завідувач кафедри д-р техн.наук., проф.  
(науковий ступінь та вчене звання)



І.Б. Туркін

(ініціали та прізвище)

Погоджено з представником здобувачів освіти:

Представник студентського  
самоуправління від  
організації № 6



Р.В. Колоткі

(ініціали та прізвище)

## 1. Загальна інформація про викладача



Манжос Юрій БСеменович, к.т.н., доцент. З 1996 року викладає в університеті. Розробник дисциплін:

- Базы Даних;
- Надійність програмно-апаратних комплексів;
- Автоматизоване тестування програмного забезпечення.

Напрями наукових досліджень: інженерія програмного забезпечення, програмне оброблення сигналів, надійність програмного забезпечення систем реального часу.

## 2. Опис навчальної дисципліни

**Семестр, в якому викладається дисципліна** – 8 семестр.

**Обсяг дисципліни:**

кредитів ЄКТС (180 годин), у тому числі аудиторних – 60 годин, самостійної роботи здобувачів – 120 годин.

**Форми здобуття освіти**

Денна, дистанційна, дуальна.

**Дисципліна** – обов'язкова.

**Види навчальної діяльності** – лекції, практичні роботи, самостійна робота здобувача.

**Види контролю** – поточний, модульний та підсумковий (семестровий) контроль (іспит).

**Мова викладання** – українська.

**Необхідні обов'язкові попередні дисципліни (пререквізити)** –

«Якість програмного забезпечення та тестування».

**Необхідні обов'язкові супутні дисципліни (кореквізити)** – «Кваліфікаційна робота».

### **3. Мета та завдання навчальної дисципліни**

**Мета:** оволодіння професійними і особистісними компетентностями з застосування сучасних систем автоматизованого тестування за рахунок сполучення таких форм навчання, як лекції, практичні роботи, виконання індивідуального завдання, а також самостійної роботи студентів.

**Завдання:** опанування студентами практичними навичками використання сучасних інтегрованих середовищ тестування.

Згідно з вимогами освітньо-професійної програми студенти повинні досягти таких компетентностей:

#### **Загальні компетентності:**

ЗК01. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК02. Здатність застосовувати знання у практичних ситуаціях.

ЗК03. Здатність спілкуватися державною мовою як усно, так і письмово.

ЗК04. Здатність спілкуватися іноземною мовою як усно, так і письмово.

ЗК05. Здатність вчитися і оволодівати сучасними знаннями.

ЗК06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК07. Здатність працювати в команді.

ЗК09. Прагнення до збереження навколишнього середовища.

ЗК12. Здатність зберігати та примножувати моральні, культурні, наукові цінності і досягнення суспільства на основі розуміння історії та закономірностей розвитку предметної області, її місця у загальній системі знань про природу і суспільство та у розвитку суспільства, техніки і технологій, використовувати різні види та форми рухової активності для активного відпочинку та ведення здорового способу життя.

#### **Фахові компетентності:**

ФК04. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами.

ФК05. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу.

ФК06. Здатність аналізувати, вибирати і застосовувати методи і засоби для забезпечення інформаційної безпеки (в тому числі кібербезпеки).

ФК08. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.

ФК10. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.

ФК11. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.

ФК13. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

ФК14. Здатність до алгоритмічного та логічного мислення.

### **Програмні результати навчання:**

ПРН01. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.

ПРН03. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.

ПРН04. Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення.

ПРН05. Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.

ПРН07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

ПРН08. Вміти розробляти людино-машинний інтерфейс.

ПРН14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.

ПРН19. Знати та вміти застосовувати методи верифікації та валідації програмного забезпечення.

ПРН20. Знати підходи щодо оцінки та забезпечення якості програмного забезпечення.

ПРН26. Вміти конфігурувати компоненти, організувати робочий процес розгортання та тестувати рішення в його остаточному операційному середовищі.

## **4. Зміст навчальної дисципліни**

### **Модуль 1.**

#### **Змістовий модуль 1. Основні поняття тестування**

##### **Тема 1. Основні поняття та методи тестування**

Форма занять: лекція, самостійна робота.

Обсяг аудиторного навантаження: 2 години.

Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): немає.

Термінологія тестування, розбіжності тестування та налагодження Тестування ПЗ Software testing - процес аналізування ПЗ та відповідної документації для вивлення дефектів та підвищення якості продукту. Дефект Defect, anomaly - відхилення фактичного результату від очікувань спостерігача, сформованих на підставі вимог, специфікацій та іншої документації. Метод білої скрині White box testing. Метод сірої скрині Gray box. Метод чорної скрині Black box testing

- *Обсяг самостійної роботи здобувачів: 5 годин.*

Опрацювання матеріалу лекцій. Формування питань до викладача.

## **Тема 2. Види тестування**

Форма занять: лекція, самостійна робота.

Обсяг аудиторного навантаження: 2 години.

Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): немає.

Ручне тестування Manual testing. Альфа тестування Alpha testing. Бета тестування Beta testing Тестирование. Динамічне тестування Dynamic testing. Димове тестування Smoke test. Інспекція (аудит) коду Code review, code inspection.. Негативне тестування Negative testing. Позитивне тестування Positive testing . Розширене тестування Extended test.

- *Обсяг самостійної роботи здобувачів: 5 годин.*

Опрацювання матеріалу лекцій. Формування питань до викладача.

## **Тема 3. Моделі розроблення ПЗ.**

Форма занять: лекція, самостійна робота.

Обсяг аудиторного навантаження: 2 години.

Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): немає.

Модель розроблення ПЗ Soft ware Development Model - структура, що систематизує різноманітні види проектної діяльності, їх взаємодію та послідовність під час розроблення ПЗ. Вибір моделі в залежності від масштабу та складності проекту, предметної області, ресурсів тощо. Вимога Requirement та визначення функцій та їх умов.

- *Обсяг самостійної роботи здобувачів: 8 годин.*

Опрацювання матеріалу лекцій. Формування питань до викладача.

Оформлення практичної роботи 1 та підготовка до її здачі

## **Тема 4 Методології розроблення програм**

Форма занять: лекція, самостійна робота.

Обсяг аудиторного навантаження: 2 години.

Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): немає.

Kanban, RUP, Extreme Programing, Feature-driven Development, Test-driven Development Структурна декомпозиція Work breakdown structure, WBS - ієрархічна декомпозиція великих задач на менші задачі. Функціональна декомпозиція Functional decomposition. Функціональне тестування Functional testing - тестування орієнтоване на перевірку коректності функціональності додатку (коректність реалізації функціональних вимог) Функціональні вимоги Functional requirements. Нефункціональні вимоги Non-functional requirements.

- *Обсяг самостійної роботи здобувачів: 5 годин.*

Опрацювання матеріалу лекцій. Формування питань до викладача.

## **Тема 5. Класифікація тестування.**

Форма занять: лекція, самостійна робота.

Обсяг аудиторного навантаження: 2 години.

Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): немає.

Спрощена класифікація тестування. Детальна класифікація тестування. Класифікація по запуску коду на виконання. Класифікація по доступу до коду та архітектури ПЗ. Класифікація по ступеню автоматизації. Класифікація по рівню деталізації ПЗ. Класифікація по ступеню важливості функцій, що тестуються. По принципам роботи з ПЗ. По природі ПЗ. По фокусуванню на рівні архітектури ПЗ. По застосуванню користувачів. По ступені формалізації. По цілям та задачам. По технікам та підходам. По моменту виконання (хронології). По приналежності до білої чи чорної скрині. Статичне тестування Static testing тестування без запуску коду. Набір тест кейсів Test case suite, test suite, test set – сукупність тест-кейсів, обраних для загальної цілі або за певною ознакою.

- *Обсяг самостійної роботи здобувачів: 5 годин.*

Опрацювання матеріалу лекцій. Формування питань до викладача.

## **Тема 6. Тестування документації та вимог**

Форма занять: лекція, самостійна робота.

Обсяг аудиторного навантаження: 2 години.

Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): немає.

Важливість вимог. Джерела та шляхи виявлення вимог. Рівні й типи вимог. Властивості якісних вимог. Приклад аналізування і й тестування вимог. Типові помилки під час аналізування та тестування вимог. Нефункціональне тестування Non-functional testing: зручність використання, сумісність, продуктивність, безпеку тощо. Аналізування першопричин Root cause analysis – процес дослідження та класифікації першопричин виникнення подій, що негативно впливають на безпеку, здоров'я, середовище, якість, надійність та виробничий процес.

- *Обсяг самостійної роботи здобувачів: 5 годин.*

Опрацювання матеріалу лекцій. Формування питань до викладача.

## **Тема 7 Статичне аналізування ПЗ.**

Форма занять: лекція, самостійна робота.

Обсяг аудиторного навантаження: 2 години.

Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): немає.

Створення моделей ПЗ. Аналізування моделей ПЗ. Переваги та недоліки статичного аналізування коду. Перевірка на рівні компіляції. Використання методів метапрограмування для статичного аналізування коду. Аналізування вхідних даних та зменшення обсягу тестів на підставі використанні розмірностей вхідних даних. Автоматичне формування програмних інваріантів та тестових випадків.

- *Обсяг самостійної роботи здобувачів: 8 годин.*

Опрацювання матеріалу лекцій. Формування питань до викладача.

Оформлення практичної роботи 2 та підготовка до її здачі

## **Тема 8. Життєвий цикл тестування ПЗ**

Форма занять: лекція, самостійна робота.

Обсяг аудиторного навантаження: 2 години.

Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): немає.

Модульне (компонентне) тестування Unit testing, component testing. Інтеграційне тестування Integration testing. Регресивне тестування Regression testing. Системне тестування System testing Тестування продуктивності Performance testing. Приймальне тестування Acceptance testing – формалізоване тестування, для перевірки ПЗ з боку користувача/замовника та формування рішення про прийом замовником роботи у виконавця.

- *Обсяг самостійної роботи здобувачів: 5 годин.*

Опрацювання матеріалу лекцій. Формування питань до викладача.

### **Модульний контроль 1**

- *Форма занять: написання модульної роботи в аудиторії (за рішенням лектора допускається проведення у дистанційній формі).*

- *Обсяг аудиторного навантаження: за необхідністю*

- *Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): немає.*

- *Обсяг самостійної роботи здобувачів – 5 годин.*

Підготовка до модульного контролю.



## **Модуль 2.**

### **Змістовий модуль 2. Автоматизоване тестування**

#### **Тема 9. Тестова документація.**

Форма занять: лекція, самостійна робота.

Обсяг аудиторного навантаження: 2 години.

Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): немає.

Проектна документація. Контрольні аркуші, тестові випадки, набори тестових випадків. Контрольний аркуш. Тестовий випадок. Атрибути тестового випадку. Інструментальні засоби керування тестуванням. Властивості якісних тестових випадків. Набори тестових випадків. Створення ефективних перевірок Типові помилки розроблення контрольних аркушів, тестових випадків та наборів тестових випадків

- *Обсяг самостійної роботи здобувачів: 1 годин.*

Опрацювання матеріалу лекцій. Формування питань до викладача.

#### **Тема 10. Звіти про дефекти**

Форма занять: лекція, самостійна робота.

Обсяг аудиторного навантаження: 2 години.

Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): немає.

Звіти про дефекти. Помилки, дефекти, збої, відмови, тощо. Звіт про дефект та його життєвий цикл. Атрибути (поля) звіту про дефект. Інструментальні засоби керування звітами про дефекти. Властивості, притаманні якісним звітам про дефекти. Створення ефективних звітів про дефекти. Типові помилки під час написання звітів про дефекти

- *Обсяг самостійної роботи здобувачів: 5 годин.*

Опрацювання матеріалу лекцій. Формування питань до викладача.

#### **Тема 11. Оцінювання витрат, планування, звітність тестування**

Форма занять: лекція, самостійна робота.

Обсяг аудиторного навантаження: 2 години.

Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): немає.

Тестовий план та звіт про тестування. Оцінювання витрат. Звіт про дефект Defect report. Звіт про результати тестування Test progress report, test summary report. Звітність Reporting. Планування Planning. Тест-план Test plan. Працевитрати Man-hours кількість робочого часу, необхідного для виконання роботи (вимірюється у людино-годинах).

- *Обсяг самостійної роботи здобувачів: 8 годин.*

Опрацювання матеріалу лекцій. Формування питань до викладача.

Оформлення практичної роботи 3 та підготовка до її здачі

### **Тема 12. Тестове покриття.**

Форма занять: лекція, самостійна робота.

Обсяг аудиторного навантаження: 2 години.

Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): немає.

Класи еквівалентності та граничні значення. Застосування в реальному житті. Клас еквівалентності Equivalence class. Гранична умова Border condition, boundary condition – значення, що знаходиться на межі класів еквівалентності. Метрика Metric - чисельна характеристика якості ПЗ. Покриття Coverage відсотковий вираз задіяння частки ПЗ відповідними наборами тестових випадків.

- *Обсяг самостійної роботи здобувачів: 5 годин.*

Опрацювання матеріалу лекцій. Формування питань до викладача.

### **Тема 13. Техніки тестування.**

Форма занять: лекція, самостійна робота.

Обсяг аудиторного навантаження: 2 години.

Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): немає.

Позитивні та негативні тестові випадки. Класи еквівалентності та граничні умови. Доменне тестування та комбінування параметрів. Попарне тестування та пошук комбінацій, що призводять до помилок. Досліджувальне тестування. Пошук причин виникнення дефектів. Доменне тестування (domain testing, domain analysis). Інструмент доменного тестування - визначення класів еквівалентності і граничних умов. Попарне тестування (pairwise testing) — перевіряються тільки комбінації кожної пари значень. Аналіз першопричин (root cause analysis) — процес дослідження та класифікації першопричин подій, що негативно впливають на безпеку, здоров'я, середовище, якість, надійність та виробничий процес.

- *Обсяг самостійної роботи здобувачів: годин.*

Опрацювання матеріалу лекцій. Формування питань до викладача.

#### **Тема 14. Автоматизація тестування.**

Форма занять: лекція, самостійна робота.

Обсяг аудиторного навантаження: 2 години.

Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): немає.

Вигоди та ризики автоматизованого тестування Automated testing – набору методів, підходів та інструментальних засобів, що дозволяє вилучити людину від виконання окремих завдань у процесі тестування.

Переваги та недоліки АТ: швидкість виконання тестів; відсутність впливу людського фактору; виконання тест-кейсів, непідвладних для людини через їх складність, швидкість тощо; можливість дослідження продуктивності ПЗ. Недоліки: значні інвестиції, збільшення проектних ризиків. Ефективність автоматизації.

Прискорення виконання тестування. Звільнення людських ресурсів для інтелектуальної роботи. Збільшення тестового покриття. Покращення коду за рахунок збільшення тестового та використання спеціальних технік автоматизації.

- *Обсяг самостійної роботи здобувачів: 5 годин.*

Опрацювання матеріалу лекцій. Формування питань до викладача.

#### **Тема 15. Особливості автоматизованого тестування.**

Форма занять: лекція, самостійна робота.

Обсяг аудиторного навантаження: 2 години.

Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): немає.

Необхідні знання та навички. Особливості тестових випадків в автоматизації. Технології **автоматизованого тестування**. Функціональна декомпозиція. Тест Test - набір з одного чи кількох тест-кейсів. Тестовий випадок Test case – набір вхідних даних, умов виконання та результату, що очікується, розроблений для перевірки властивості чи поведінки ПЗ. Тест-кейс – документ - формальний запис тест-кейса.

- *Обсяг самостійної роботи здобувачів: 8 годин.*

Опрацювання матеріалу лекцій. Формування питань до викладача.

Оформлення лабораторної роботи 4 та підготовка до її здачі

## **Тема 16. Особливості автоматизованого тестування критичного шляху**

Форма занять: лекція, самостійна робота.

Обсяг аудиторного навантаження: 2 години.

Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): немає.

Critical path test -дослідження функціональності типового користувача у типовій повсякденній діяльності. Тестування під керуванням даними Data-driven testing. Тестування за ключовими словами Keyword-driven testing. Тестування під керуванням поведінки Behavior driven testing. Використання фреймворків для автоматизації тестування. Технологія запису та відтворення (Record & Playback). Навантажувальне тестування.

- *Обсяг самостійної роботи здобувачів: 5 годин.*

Опрацювання матеріалу лекцій. Формування питань до викладача.

## **Тема 17 Сучасні засоби автоматизованого тестування.**

Форма занять: лекція, самостійна робота.

Обсяг аудиторного навантаження: 2 години.

Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): немає.

Інструментальні засоби JUnit та NUnit. Інсталяція засобу у MS VS. Основні ключові слова та інтерфейс API. Реалізація модульного, інтеграційного, системного та регресійного тестування.

- *Обсяг самостійної роботи здобувачів: 5 годин.*

Опрацювання матеріалу лекцій. Формування питань до викладача.

## **Тема 18 Сучасні фреймворки автоматизованого тестування**

Форма занять: лекція, самостійна робота.

Обсяг аудиторного навантаження: 2 години.

Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): немає.

Selenium фреймворк з відкритим кодом для автоматизації тестування веб-додатків. Написання скриптів для нього. Katalon Studio - інструмент автоматизації процесу тестування веб-додатків, мобільних додатків і веб-сервісів. Unified Functional Testing або UFT комерційний інструмент функціонального тестування графічного інтерфейсу десктопних, мобільних і веб-додатків на всіх існуючих платформах. Watir – це інструмент з відкритим кодом автоматизації тестування веб-додатків. IBM Rational Functional Tester – платформа для керованого функціонального та регресійного тестування ПЗ. Ranorex – платний універсальний інструмент для автоматизації тестування веб-, мобільних та десктопних додатків. Перспективи автоматизації тестування

- *Обсяг самостійної роботи здобувачів: 8 годин.*

Опрацювання матеріалу лекцій. Формування питань до викладача.

Оформлення практичної роботи 5 та підготовка до її здачі

Оформлення індивідуального завдання та підготовка до його здачі

## **Модульний контроль 2**

- *Форма занять: написання модульної роботи в аудиторії (за рішенням лектора допускається проведення у дистанційній формі).*

- *Обсяг аудиторного навантаження: за необхідністю*

- *Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): немає.*

- *Обсяг самостійної роботи здобувачів – 5 годин.*

Підготовка до модульного контролю.

## **5. Індивідуальні завдання**

Індивідуальне завдання полягає у розробленні тестового плану, виконання автоматизованого тестування та оформленні тестової документації, що стосується програмного забезпечення кваліфікаційної роботи

## **6. Методи навчання**

Словесні, наочні, практичні.

## **7. Методи контролю**

Поточний контроль (теоретичне опитування й розв'язання практичних завдань), модульний контроль (тестування за розділами курсу) та підсумковий (семестровий) контроль (іспит).

## **8. Критерії оцінювання та розподіл балів, які отримують здобувачі**

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
<b>Змістовний модуль 1</b>			
Виконання і захист практичних робіт	1...5	2	2...10
Модульний контроль	2...25	1	2...25
<b>Змістовний модуль 2</b>			
Виконання і захист практичних робіт	1...5	3	3...15
Модульний контроль	5...25	1	5...25
Виконання і захист індивідуального завдання	2...25	1	2...25
<b>Усього за семестр</b>			<b>14...100</b>

## Прийнята шкала оцінювання

Сума балів за всі види навчальної діяльності	Оцінка для екзамену
90-100	відмінно
75-89	добре
60-74	задовільно
01-59	незадовільно з можливістю повторного складання

Семестровий контроль (іспит) проводиться у разі відмови студента від балів поточного тестування й за наявності допуску до іспиту. Під час складання семестрового іспиту студент має можливість отримати максимум 50 балів, які замінюють результати п'яти модульних контролів.

Білет для іспиту складається з двох теоретичних питань (кожне питання 33 бали) та одного практичного питання (питання 34 балів).

### Критерії оцінювання роботи здобувача протягом семестру

**Задовільно (60-74).** Показати мінімум знань та умінь. Знати основні поняття та положення та етапи тестування програмного забезпечення.

**Добре (75-89).** Твердо знати мінімум, виконати всі контрольні роботи та поза аудиторну самостійну роботу. Досконало знати основні поняття та положення теорії верифікації та валідації, методи тестування, вміти проектувати тестові випадки, використовувати класи еквівалентності та доменний аналіз для ефективного виконання тестів.

Досконало володіти інструментальними засобами розроблення та супроводу автоматизованого тестування ПЗ. Вміти створювати тестові випадки та реалізовувати їх у реальних проектах.

**Відмінно (90-100).** Здати всі контрольні роботи та індивідуальне завдання (розрахункову роботу) з оцінкою «відмінно». Досконало знати всі теоретичні питання, вміти розв'язувати завдання за темами, які вивчалися, вміти застосовувати набуті теоретичні знання та практичні навички в розробленні алгоритмів та програм автоматизованого тестування програмного забезпечення.

## 9. Політика навчального курсу

Відпрацювання пропущених занять відбувається відповідно до розкладу консультацій, за попереднім погодженням з викладачем. Питання, що стосуються академічної доброчесності, розглядає викладач або за процедурою, визначеною у Положенні про академічну доброчесність.

## 10. Методичне забезпечення та інформаційні ресурси

Підручники, навчальні посібники, навчально-методичні посібники, конспекти лекцій, методичні рекомендації з проведення лабораторних робіт тощо, які видані в Університеті знаходяться за посиланням:

1. Дистанційний курс дисципліни розроблено у системі дистанційного навчання Mentor, яку впроваджено в Національному аерокосмічному університеті ім. М.Є. Жуковського «ХАІ», доступ до курсу «Автоматизоване тестування програмного забезпечення» за посиланням: <https://mentor.khai.edu/course/view.php?id=4866>
2. «Автоматизоване тестування програмного забезпечення. Конспект лекцій». Навчальний посібник по лабораторному практикуму» розроблено у системі дистанційного навчання Mentor, яку впроваджено в Національному аерокосмічному університеті ім. М.Є. Жуковського «ХАІ», доступ до посібника за посиланням: <https://mentor.khai.edu/course/view.php?id=4866>
3. «Автоматизоване тестування програмного забезпечення. Конспект лекцій». Навчальний посібник з виконання практичних робіт» розроблено у системі дистанційного навчання Mentor, яку впроваджено в Національному аерокосмічному університеті ім. М.Є. Жуковського «ХАІ», доступ до посібника за посиланням: <https://mentor.khai.edu/course/view.php?id=4866>

## 11. Рекомендована література

### Базова

1. Білас О. Є. Якість програмного забезпечення та тестування. Навчальний посібник. Львів : Львівська політехніка, 2011. – 216 с.
2. Джон Макгрегор, Девід Сайкс Тестування об'єктно-орієнтованого програмного забезпечення. Практичний посібник. – К.: ДіаСофт – 2002. – 432 с.
3. Unit testing C# with NUnit and .NET Core [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.microsoft.com/en-us/dotnet/core/testing/unit-testing-with-nunit>
4. What Is NUnit? [Електронний ресурс] – Режим доступу до ресурсу: <https://nunit.org>
5. JUnit Tutorial [Електронний ресурс] – Режим доступу до ресурсу: <https://www.tutorialspoint.com/junit/index.htm>

## Допоміжна

1. IEEE Guide to Software Engineering Body of Knowledge, SWEBOOK, 2004
2. Android Developers [Електронний ресурс] – Режим доступу до ресурсу: <https://developer.android.com/index.html>
3. Android Developers Blog [Електронний ресурс] – Режим доступу до ресурсу: <http://android-developers.blogspot.com/>
4. Kolawa, Adam; Huizinga, Dorota (2007). *Automated Defect Prevention: Best Practices in Software Management*. Wiley-IEEE Computer Society Press. p. 74. [ISBN 978-0-470-04212-0](#).
5. *Proceedings from the 5th International Conference on Software Testing and Validation (ICST)*. Software Competence Center Hagenberg. "Test Design: Lessons Learned and Practical Implications. [doi:10.1109/IEEESTD.2008.4578383](#). [ISBN 978-0-7381-5746-7](#).
6. Brian Marick. "[When Should a Test Be Automated?](#)". *StickyMinds.com*. Retrieved 2009-08-20.
7. Garousi, Vahid; Mäntylä, Mika V. (2016-08-01). "When and what to automate in software testing? A multi-vocal literature review". *Information and Software Technology*. **76**: 92–117. [doi:10.1016/j.infsof.2016.04.015](#).
8. Vodde, Bas; Koskela, Lasse (2007). "Learning Test-Driven Development by Counting Lines". *IEEE Software*. **24** (3): 74–79. [doi:10.1109/ms.2007.80](#). [S2CID 30671391](#).
9. [Testing APIs protects applications and reputations](#), by Amy Reichert, SearchSoftwareQuality March 2015
10. [All About API Testing: An Interview with Jonathan Cooper](#), by Cameron Philipp-Edmonds, StickyMinds August 19, 2014
11. [Produce Better Software by Using a Layered Testing Strategy](#), by Sean Kenefick, [Gartner](#) January 7, 2014
12. [Part of the Pipeline: Why Continuous Testing Is Essential](#), by Adam Auerbach, TechWell Insights August 2015
13. Headless Testing with Browsers; <https://docs.travis-ci.com/user/gui-and-headless-browsers/>
14. Headless Testing with PhantomJS; <http://phantomjs.org/headless-testing.html>
15. Automated User Interface Testing; <https://www.devbridge.com/articles/automated-user-interface-testing/>
16. [Jump up to: Mike Cohn \(2010\). Succeeding with Agile](#). Raina Chrobak. [ISBN 978-0-321-57936-2](#).
17. [The Practical Test Pyramid](#), by Ham Vocke
18. "[Selenium Meet-Up 4/20/2010 Elisabeth Hendrickson on Robot Framework 1of2](#)". Retrieved 2010-09-26.