

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

Кафедра математичного моделювання та штучного інтелекту (№ 304)

ЗАТВЕРДЖУЮ

Голова НМК 2



(підпис)

Д. М. Крицький

(ініціали та прізвище)

« 31 » серпня 2021 р.

**СИЛАБУС ВИБІРКОВОЇ
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Автоматизоване тестування

(назва навчальної дисципліни)

Галузь знань: 10 Природничі науки, 11 Математика та статистика, 12 Інформаційні технології, 15 Автоматизація та приладобудування, 16 Хімічна та біоінженерія, 17 Електроніка та телекомунікації, 19 Архітектура та будівництво

Спеціальність: 101 Екологія, 103 Науки про Землю, 113 Прикладна математика, 121 Інженерія програмного забезпечення, 122 Комп'ютерні науки, 123 Комп'ютерна інженерія, 124 Системний аналіз, 125 Кібербезпека, 151 Автоматизація та комп'ютерно-інтегровані технології, 152 Метрологія та інформаційно-вимірювальна техніка, 163 Біомедична інженерія, 172 Телекомунікації та радіотехніка, 173 Авіоніка, 193 Геодезія та землеустрій

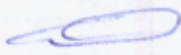
Освітня програма: усі освітні програми за відповідними спеціальностями

Рівень вищої освіти: *другий (магістерський)*

Силабус введено в дію з 01.09.2021 року

Харків – 2021 р.

Розробник: Меняйлов Є. С., ст. викл.
(прізвище та ініціали, посада, науковий ступінь і вчене звання)


(підпис)

Силабус навчальної дисципліни розглянуто на засіданні кафедри
математичного моделювання та штучного інтелекту
(назва кафедри)

Протокол № 2 від « 27 » 08 2021 р.

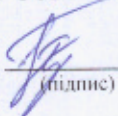
Завідувач кафедри д.т.н. доцент
(науковий ступінь та вчене звання)


(підпис)

Андрій ЧУХРАЙ
(ім'я та прізвище)

Погоджено з представником здобувачів освіти:

Голова студентської самоврядування
факультету №3


(підпис)

А.І.Белашов
(ім'я та прізвище)

1. Загальна інформація про викладача



Меняйлов Євген Сергійович. З 2015 року викладає в університеті наступні дисципліни:

- архітектура обчислювальних систем;
- проектування програмних систем;
- інтелектуальні системи;
- аналіз даних.

Напрями наукових досліджень:

1. Математичні моделі і методи розв'язання задач вдосконалення технічних систем в умовах невизначеності вхідних даних.

2. Інтелектуальні моделі, методи та інформаційні технології для дослідження статистичних даних о пацієнтах в медичній діагностиці.

2. Опис навчальної дисципліни

Семестр, в якому викладається дисципліна – 1 семестр.

Обсяг дисципліни: 5 кредитів ЄКТС / 150 годин, у тому числі аудиторних – 64 год., самостійної роботи здобувачів – 86 год.

Форма здобуття освіти – денна, дистанційна, дуальна.

Дисципліна вибіркова.

Види навчальної діяльності – лекції, практичні (лабораторні) заняття

Види контролю – модульний контроль, іспит.

Мова викладання – українська.

Пререквізити – програмування, знання з об'єктно орієнтованого програмування, володіння навичками створення та користування діаграмами на мові UML.

Кореквізити – розробника програмного забезпечення, інтелектуальні системи, інтелектуальні програмні комплекси, тестування, project management, написання магістерської роботи.

3. Мета та завдання навчальної дисципліни

Мета: вивчення навчальної дисципліни «Автоматизоване тестування» полягає в формуванні системи знань, способів діяльності та творчих здібностей з основних теоретичних положень про підходи до автоматизованого тестування програмних продуктів через використання різноманітних підходів, а також засвоєння вмінь які б дозволяли на практиці реалізувати ці знання.

Завдання: засвоєння знань про особливості тестування програмних продуктів; вивчення підходів (видів тестування) та засобів (програмних бібліотек) для автоматизованого тестування.

Компетентності, які набуваються:

- здатність проводити тестування елементи програмної архітектури сучасних інформаційних систем;
- здатність проводити тестування функціональних модулів програмних систем;
- здатність використовувати уніфіковану мову проектування UML для опису загальної архітектури складних систем, або відповідних її елементів;
- здатність розроблювати програмне забезпечення відповідно до обраної моделі життєвого циклу інформаційних систем;
- здатність використовувати різні підходи до тестування програмних систем та їх компонентів;
- здатність використовувати бібліотеки для GUI-автоматизації.

Очікувані результати навчання:

- вміти створювати план тестування програмного продукту або функціональних елементів програмного продукту;
- обирати належні засоби для розробки або дослідження (середовище розробки, мова програмування, програмне забезпечення та програмні пакети тощо), що дозволяють знайти правильне і ефективне рішення;
- мати уявлення про види тестування програмних систем, та практичні аспекти їх використання;
- знати відповідну до предметної області термінологію;
- знати особливості проведення тестування функціональності і поведінки складної системи.

4. Зміст навчальної дисципліни

Змістовний модуль 1. Види та підходи до тестування.

- *Обсяг самостійної роботи здобувачів: 12-16 годин – самостійне опрацювання матеріалу за темами.*

Інші види тестування: тестування білого ящика, тестування чорного ящика, тестування сірого ящика, альфа та бета тестування, тестування продуктивності (навантажувальне), тестування надійності. Підходи до тестування за природою об'єкта тестування: системи реального часу (real-time systems), web додатки, e-business systems, наукові.

Тема 1. Види та підходи до тестування.

- *Форма занять: лекція, самостійна робота.*
- *Обсяг аудиторного навантаження: 12-14 годин.*
- *Практична (лабораторна) робота: «Розробка плану тестування».*
- *Практична (лабораторна) робота: «Ручне тестування програмного продукту та створення звіту про дефекти».*

Bug Tracking System – системи реєстрації і відстеження життєвого циклу дефектів (помилки). План тестування (test plan) з варіантами тестування (test cases). Bug Report – звіт про дефекти в програмі. Класи помилки.

Класифікація методів тестування. Види тестування: ручне тестування (manual testing), модульне тестування (unit testing), інтеграційне тестування (integration testing), приймальне тестування (acceptance testing), випускне тестування (release testing).

- *Обсяг самостійної роботи здобувачів: 1-2 години.*

Опрацювання матеріалу лекцій. Формування питань до викладача. Оформлення практичної (лабораторної роботи) та підготовка до її здачі.

Модульний контроль

- *Форма занять: написання модульної роботи в аудиторії (за рішенням лектора допускається проведення у дистанційній формі).*
- *Обсяг аудиторного навантаження: 2 години*
- *Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): відсутні.*
- *Обсяг самостійної роботи здобувачів – за необхідністю. Підготовка до модульного контролю.*

Змістовний модуль 2. Підходи та засоби автоматизованого тестування.

- *Обсяг самостійної роботи здобувачів: 12-16 годин – самостійне опрацювання матеріалу за темами.*

Бібліотеки для автоматизованого тестування: xUnit – тестування додатків для .NET; TestNG – тестування додатків для Java; WatiN – тестування веб-додатків з підтримкою браузерів Internet Explorer, Mozilla Firefox; UniTESK – тестування додатків на Java, C.

Тема 1. Підходи до автоматизованого тестування.

- Форма занять: лекція, самостійна робота.
- Обсяг аудиторного навантаження: 4-6 годин.

Основні підходи до автоматизації тестування: тестування на рівні коду і GUI-тестування. Огляд інструментів для автоматизованого тестування.

GUI-автоматизація: утиліти запису і відтворення (capture/playback tools), сценарії (scripting), data-driven testing.

- Обсяг самостійної роботи здобувачів: 1-2 години.

Опрацювання матеріалу лекцій. Формування питань до викладача. Оформлення практичної (лабораторної роботи) та підготовка до її здачі.

Тема 2. Інструментарій для автоматизованого тестування.

- Форма занять: лекція, практична (лабораторна) робота, самостійна робота.
- Обсяг аудиторного навантаження: 12-14 годин.
- Практична (лабораторна) робота: «Використання інструментарію бібліотеки JUnit для тестування додатків для Java».
- Практична (лабораторна) робота: «Використання інструментарію NUnit відкритого середовища модульного тестування застосунків для .NET».
- Практична (лабораторна) робота: «Використання інструменту Selenium для автоматизації роботи в web-браузері».
- Практична (лабораторна) робота: «Використання середовища TOSCA TestSuite для автоматичного тестування додатків для HTML, Java, .NET».

Бібліотека JUnit для тестування програмного забезпечення для мови Java та її компоненти: junit.framework.Assert (assertEquals, assertFalse, assertNotNull, assertNull, assertNotSame, assertEquals, assertTrue), junit.framework.TestCase extends junit.framework.Assert (run, setUp, tearDown).

NUnit – відкрите середовище модульного тестування застосунків для .NET, перенесене з мови Java (бібліотека JUnit) та його розширення оригінального пакету NUnit, значна частина з них також з відкритим кодом: FireBenchmarks, NUnit.Forms, NUnit.ASP.

Selenium – інструмент для автоматизації роботи в web-браузері. Selenium, як спосіб створювати тести вебзастосунків без вивчення мов програмування. Інструмент має власну предметно-орієнтовану мову (Selenese) задля написання тестів на таких мовах як C#, Groovy, Java, Perl, PHP, Python, Ruby та Scala.

Компоненти: Selenium IDE, Selenium Client API, Selenium Remote Control, Selenium WebDriver, Selenium Grid.

TOSCA TestSuite – середовище автоматичного тестування додатків для HTML, Java, .NET, написане на мовах VB6, Java і C#. Програма орієнтована на тестування без знань програмування. Готові конфігурації для тестування бізнес-додатків SAP і Siebel CRM. Використання Microsoft SQL Server, Oracle Database або DB2 для зберігання інфраструктури та підтримки спільної роботи.

- *Обсяг самостійної роботи здобувачів: 1-2 години.*

Опрацювання матеріалу лекцій. Формування питань до викладача. Оформлення практичної (лабораторної роботи) та підготовка до її здачі.

Модульний контроль.

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

- *Обсяг аудиторного навантаження: 2 години*

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

- *Обсяг самостійної роботи здобувачів – за необхідністю. Підготовка до модульного контролю.*

5. Індивідуальні завдання

Не передбачено навчальним планом

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
Змістовний модуль 1			
Робота на лекціях	0...2	5	0...10
Виконання і захист лабораторних (практичних) робіт	0...10	2	0...20
Модульний контроль	0...15	1	0...20

Змістовний модуль 2			
Робота на лекціях	0...2	5	0...10
Виконання і захист лабораторних (практичних) робіт	0...5	4	0...20
Модульний контроль	0...15	1	0...20
Усього за семестр			0...100

Іспит проводиться у вигляді тестування та виконання практичних завдань. Тест складається з 20 питань закритого типу (за правильну відповідь на одне питання здобувач отримує 3 бала) та двох практичних завдань (максимальна кількість балів за практичне завдання – 20).

Під час складання семестрового іспиту здобувач має можливість отримати максимум 100 балів.

Критерії оцінювання роботи здобувача протягом семестру

Задовільно (60-74) – мати мінімум знань і умінь для забезпечення програмних результатів навчання. Відпрацювати та захистити всі практичні роботи.

Добре (75-89) – знати основні теми дисципліни. Достатньо знати основні етапи створення програмних продуктів, особливості проведення тестування та види тестування. Відпрацювати та захистити всі практичні роботи.

Відмінно (90-100) – мати знання, що дозволять самостійно, вільно та обґрунтовано відповідати на будь які питання щодо проведення тестування програмних продуктів, або їх функціональних елементів, відповідно до обраного стеку технологій. Відпрацювати та захистити всі лабораторні роботи.

Шкала оцінювання: бальна і традиційна

Сума балів	Оцінка за традиційною шкалою	
	Іспит, диференційований залік	Залік
90 – 100	Відмінно	Зараховано
75 – 89	Добре	
60 – 74	Задовільно	
0 – 59	Незадовільно	Не зараховано

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

Не має.

11. Рекомендована література

Базова

1. Automated Software Engineering: A Deep Learning-Based Approach (Learning and Analytics in Intelligent Systems) by Suresh Chandra Satapathy (Author), Ajay Kumar Jena (Contributor), Jagannath Singh (Contributor), Saurabh Bilgaiyan (Contributor). Springer; 1st ed. 2020 edition (January 8, 2021), 132 p.
2. Automated Software Testing: Foundations, Applications and Challenges (Services and Business Process Reengineering) by Ajay Kumar Jena (Editor), Himansu Das (Editor), Durga Prasad Mohapatra (Editor). Springer; 1st ed. 2020 edition (February 4, 2021). 180 p.
3. Essential Logic for Computer Science. Illustrated Edition by Rex Page (Author), Ruben Gamboa (Author). The MIT Press; Illustrated edition (January 8, 2019). 304 p.
4. Practical Test Design: Selection of traditional and automated test design techniques Illustrated Edition. BCS, The Chartered Institute for IT; Illustrated edition (August 28, 2019). 360 p.
5. Practical Security Automation and Testing: Tools and techniques for automated security scanning and testing in DevSecOps by Tony Hsiang-Chih Hsu. Packt Publishing (February 4, 2019). 256 p.

Допоміжна

1. Selenium Framework Design in Data-Driven Testing: Build data-driven test frameworks using Selenium WebDriver, AppiumDriver, Java, and TestNG by Carl Cocchiari. Packt Publishing (January 23, 2018). 354 p.
2. Design, Build, Ship: Faster, Safer Software Delivery by Sam Newman (Author). O'Reilly Media; 1st edition (June 4, 2021).

12. Інформаційні ресурси

1. 29 Best Automated Software Testing Books of All Time.
<https://bookauthority.org/books/best-automated-software-testing-books>
2. Automation Testing Vs Manual Testing
<https://www.softwaretestingclass.com/automation-testing-vs-manual-testing/>