

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ «МЕТОДОЛОГІЯ РОЗРОБКИ СУЧАСНИХ ВЕБРЕСУРСІВ ТА СЕРВІСІВ»



Ступінь освіти	Доктор філософії
Галузь знань	12 Інформаційні технології
Загальний обсяг	4 кредити ЄКТС (120 годин)
Тривалість викладання	7 чверть
Заняття:	
лекції:	3 години на тиждень
практичні заняття:	2 години на тиждень
Мова викладання	українська

Сторінка курсу в СДО НТУ «ДП»:
<https://do.nmu.org.ua/course/view.php?id=4966>

Кафедра, що викладає Програмного
забезпечення
комп'ютерних систем



Викладач:
Спірінцев В'ячеслав Васильович
Доцент, к.т.н.

Персональна сторінка
<https://pzks.nmu.org.ua/ua/teachers/teachers.php>

E-mail: Spirintsev.V.V@nmu.one

1. Анотація до курсу

Зараз в сучасному світі спостерігається тенденція всебічного застосування комп'ютерної техніки та технологій в різних сферах людської діяльності. Web-технології не стали виключенням. Це обумовлено появою та розвитком глобальної телекомунікаційної мережі Internet, що надає якісно новий рівень засобів зв'язку та глобалізацію інформаційного простору. Основною метою курсу є формування бази теоретичних та практичних знань в області розробки сучасних веб-ресурсів та сервісів для здобувачів освіти Національного технічного університету «Дніпровська політехніка» спеціальності 122 «Комп'ютерні науки» для вирішення різноманітних практичних задач у їхній діяльності, що сприяє підвищенню якості їхньої підготовки. В рамках курсу розглядаються основні моделі та методології розробки сучасних веб-ресурсів та сервісів; сучасні тенденції розвитку клієнтських та серверних технологій веб-програмування; технології, архітектурні стилі та безпека веб-сервісів; основні

принципи розробки програмного забезпечення та дизайн-патерни; основні поняття, цілі, проблеми, принципи та прийоми рефакторингу програмного забезпечення; основні підходи до методів та засобів тестування програмних систем, принципи побудови та функціонування процесу тестування програмного забезпечення, управління вимогами в програмних системах; основні поняття та принципи хмарних технологій, веб-служби, що надаються концепцією хмарних обчислень; питання безпеки веб-ресурсів, загальними сценаріями основних атак на веб-застосунки та уразливими містами; питання оптимізації веб-застосувань.

2. Мета та завдання курсу

Мета дисципліни – формування у здобувачів освіти глибокого розуміння концепції та формування системи компетенцій в питаннях методології розробки сучасних веб-ресурсів та сервісів, основ технології Web-сервісів у специфікаціях першого та другого рівня та реалізації технології Web-сервісів у вигляді стандартів платформи Java; принципами розробки та архітектури програмного забезпечення; знайомство здобувачів освіти з сучасними тенденціями розвитку клієнтських та серверних технологій веб-програмування, веб-сервісів; подальше становлення і вдосконалення інформаційної та програмної культури майбутніх фахівців.

Завданнями дисципліни є:

- опанування теоретико-понятійної бази курсу;
- ознайомлення здобувачів освіти з моделями і методологіями розробки сучасних веб-ресурсів та сервісів;
- ознайомлення здобувачів освіти з основними поняттями, цілями, проблемами, принципами та прийомами рефакторингу програмного забезпечення;
- ознайомлення здобувачів освіти з основними патернами проектування;
- ознайомлення здобувачів освіти з сучасними тенденціями розвитку клієнтських та серверних технологій веб-програмування, веб-сервісів, хмарних технологій та веб-служб, що надаються концепцією хмарних обчислень;
- ознайомлення здобувачів освіти з питаннями безпеки веб-ресурсів, загальними сценаріями основних атак на веб-застосунки та уразливими містами.

3. Дисциплінарні результати навчання

Дисциплінарні результати навчання:

- знати сучасні тенденції розвитку клієнтських та серверних технологій веб-програмування, веб-сервісів та веб-служб, методи та інструментальні засоби для проектування і створення веб-застосувань;

- аналізувати та мотивовано обирати оптимальну методологію, інструментальні засоби та технології розробки для розв’язання завдань створення і супроводження сучасних веб-ресурсів та сервісів;
- знати основні принципи розробки програмного забезпечення та дизайн-патерни, прийоми рефакторингу, основні підходи до методів та засобів тестування програмних систем, принципи побудови та функціонування процесу тестування програмного забезпечення, управляти вимогами в програмних системах та застосовувати їх при розробці;
- знати основи технології Web-сервісів у специфікаціях першого та другого рівня та реалізації технології Web-сервісів у вигляді стандартів платформи Java;

4. Структура курсу

Види та тематика навчальних занять	Внесок в загальну оцінку, %
ЛЕКЦІЇ	40
Тема 1. Введення. Принципи розробки та архітектури програмного забезпечення Основні принципи проектування ПЗ. DRY. KISS. YAGNI. SOLID. Дизайн-патерни. Класифікація патернів. Породжуючі патерни. Структурні патерни. Патерни поведінки. Архітектура веб-застосунків. Тестування. Основні принципи тестування. Фундаментальний процес тестування (ISTQB). Тестова документація.	
Тема 2. Рефакторинг програмного забезпечення Поняття та принципи рефакторингу. Основні цілі та причини проведення рефакторингу. Проблеми рефакторингу. Code Smells (запахи коду). Прийоми рефакторингу: складання методів, переміщення функцій між об'єктами, організація даних, спрощення умовних виразів, спрощення викликів методів, рішення завдань узагальнення. Інструментальні засоби проведення рефакторингу.	
Тема 3. Моделі життєвого циклу програмного забезпечення Code and fix, Waterfall Model, V-model, Incremental Model, Iterative Model, Spiral Model, RAD-Model.	
Тема 4. Методологія розробки сучасних веб-ресурсів та сервісів Гнучка методологія Agile. Принципи Agile. Kanban, Scrum, XP, Lean, FDD, TDD, Cleanroom Software Engineering, OpenUP, MSF, DSDM, Crystal Clear, Agile Unified Process, ICONIX.	
Тема 5. Управління вимогами в програмних системах Класифікація вимог. Визначення вимог до якості програмного забезпечення та план забезпечення якості програмного забезпечення. Smart (Specific, Measurable, Achievable, Relevant, Time bound). Пріоритезація вимог. Властивості якісних вимог. Інженерія вимог. Відстеження вимог протягом життєвого циклу програмного забезпечення. Управління вимогами Agile. Фактори якості програмного забезпечення. Проблеми якості програмного забезпечення. Методології підвищення якості програмних систем. Критерії успіху.	
Тема 6. Технології розробки сучасних веб-ресурсів та серверних веб-застосунків HTML, DHTML, CSS, XML, XSLT, DOM, JavaScript. Серверні мови	

Види та тематика навчальних занять	Внесок в загальну оцінку, %
програмування. PHP, ASP, MySQL, Ajax. Бібліотеки. Фреймворки. Інструменти. CSS-фреймворк Bootstrap. JS- бібліотеки і фреймворки: jQuery, Angular.js, React.js, Vue.js, Ember.js, Meteor.js. Огляд різних редакторів коду та IDE для розглянутих технологій. Інтерфейс CGI, ISAPI. Сценарії. ASP. Архітектура веб-застосунків ASP.NET. Розробка веб-застосунків на платформі .NET. Серверні елементи управління ASP.NET. Архітектура MVC.	
Тема 7. Організація процесу розробки веб-контента. CMS/CMF системи. SEO. Оптимізація веб-застосувань. Система управління контентом (CMS). Огляд CMS-систем. Плагіни та шаблони для CMS. Оптимізація веб-застосувань.	
Тема 8. Веб-сервіси. Безпека веб-ресурсів. Визначення веб-сервера. Технології веб-сервісів: XML/JSON over HTTP, XML-RPC, SOAP и WSDL. Архітектурні стилі веб-сервісів: RPC, SOA, RESTful. Безпека веб-сервісів. Cross Site Scripting. SQL-Injection. Cross Site Request Forgery. Upload-уразливості. Include-уразливості. Витік інформації. Захист клієнта і сервера веб-застосувань.	
<i>Тестова контрольна робота №1 (за темами 1-8).</i>	40
ПРАКТИЧНІ ЗАНЯТТЯ	60
Практичне заняття 1. Створення JAX-WS Web-сервісу і JAX-WS-клієнту	8
Практичне заняття 2. Створення RESTful Web-сервісу засобами Java.	8
Практичне заняття 3. Створення RESTful Web-сервісу засобами Groovy & GRAILS.	8
Практичне заняття 4. Розробка корпоративного додатку EJB.	8
Практичне заняття 5. Створення корпоративного додатку з використанням Hibernate.	8
Практичне заняття 6. Створення Java EE додатка з використанням JSF 2.1, EJB 3.1, JPA.	10
Практичне заняття 7. Створення веб-сервісу, використання JavaFx на клієнтській стороні архітектури «клієнт-сервер».	10
РАЗОМ	100

5. Технічне обладнання та/або програмне забезпечення

Використовуються лабораторії кафедри програмного забезпечення комп'ютерних систем (комп'ютерне та мультимедійне обладнання).

Дистанційна платформа Moodle, MS Office 365, Microsoft Teams.

Java Development Kit (JDK) - безкоштовний розповсюджуваний Oracle комплект розробника застосунків на мові Java, який включає до себе компілятор Java (javac), стандартні бібліотеки класів Java, приклади, документацію, різноманітні утиліти (JConsole та ін.) і виконавчу систему Java (JRE). NetBeans IDE або IntelliJ IDEA – вільне інтегроване середовище розробки додатків, MySQL. Фреймворк Grails, Hibernate, платформа Java EE (Java Enterprise Edition). Всі продукти є безкоштовні, поширюються через публічну ліцензію.

6. Система оцінювання та вимоги

6.1. Навчальні досягнення здобувачів вищої освіти за результатами вивчення курсу оцінюватимуться за шкалою, що наведена нижче:

Рейтингова шкала	Інституційна шкала
90 – 100	відмінно
74-89	добре
60-73	задовільно
0-59	незадовільно

6.2. Здобувачі вищої освіти можуть отримати підсумкову оцінку з навчальної дисципліни на підставі поточного оцінювання знань за умови, якщо набрана кількість балів захисту практичних робіт складатиме не менше 60 балів.

Теоретична частина оцінюється за результатами здачі контрольної тестової роботи, що містить тестові закриті запитання з однією вірною відповіддю (максимальна кількість – 40 балів), тобто 40% від оцінки за дисципліну.

Практичні роботи (сім робіт – у вигляді індивідуального завдання з кожної, розподіл % див. в таблиці розділу 4) виконуються у письмовому вигляді (звіт з кожної роботи оцінюється в межах балів, представлених в таблиці розділу 4, загалом практичні враховуються як 60% (максимум 60 балів). Отримані бали за теоретичну частину та лабораторні роботи додаються і є підсумковою оцінкою за вивчення навчальної дисципліни. Максимально за поточною успішністю здобувач вищої освіти може набрати 100 балів.

6.3. Критерії оцінювання підсумкової роботи. У випадку якщо здобувач вищої освіти за поточною успішністю отримав менше 60 балів та/або прагне поліпшити оцінку, проводиться підсумкове оцінювання.

Підсумкове оцінювання за дисципліною проводиться у вигляді комплексної контрольної роботи, яка включає запитання з теоретичної та практичної частини курсу. Білет складається з 30 тестових завдань з чотирма варіантами відповідей, одна правильна відповідь оцінюється в 2 бали (разом 60 балів) та 1 завдання з практичної частини, що оцінюється максимум у 40 балів, причому:

- 40 балів – відповідність еталону;
- 30 балів – відповідність еталону з незначними помилками;
- 20 балів – часткова відповідність еталону, питання повністю не розкриті;
- 10 балів – невідповідність еталону, але відповідність темі запитання;
- 0 балів – відповідь не наведена або не відноситься до теми запитання.

Отримані бали за тестові завдання та завдання з практичної частини додаються і є підсумковою оцінкою за вивчення навчальної дисципліни. Максимально за підсумковою роботою здобувач вищої освіти може набрати 100 балів.

7. Політика курсу

7.1. Політика щодо академічної доброчесності. Академічна доброчесність здобувачів вищої освіти є важливою умовою для опанування результатами навчання за дисципліною і отримання задовільної оцінки з поточного та підсумкового контролів. Академічна доброчесність базується на засудженні практик списування (виконання письмових робіт із залученням зовнішніх джерел інформації, крім дозволених для використання), плагіату (відтворення опублікованих текстів інших авторів без зазначення авторства), фабрикації (вигадування даних чи фактів, що використовуються в освітньому процесі). Політика щодо академічної доброчесності регламентується положенням "Положення про систему запобігання та виявлення плагіату у Національному технічному університеті "Дніпровська політехніка" <https://cutt.ly/MCfh5kv>

У разі порушення здобувачем вищої освіти академічної доброчесності (списування, плагіат, фабрикація), робота оцінюється незадовільно та має бути виконана повторно. При цьому викладач залишає за собою право змінити тему завдання.

7.2. Комунікаційна політика. Здобувачі вищої освіти повинні мати активовану корпоративну університетську пошту.

Усі письмові запитання до викладачів стосовно курсу мають надсилатися на університетську електронну пошту.

7.3. Політика щодо перескладання. Роботи, які здаються із порушенням термінів без поважних причин оцінюються на нижчу оцінку. Перескладання підсумкового оцінювання відбувається із дозволу деканату за наявності поважних причин (наприклад, лікарняний).

7.4 Політика щодо оскарження оцінювання. Якщо здобувач вищої освіти не згоден з оцінюванням його знань він може опротестувати виставлену викладачем оцінку у встановленому порядку.

7.5. Бонуси. Здобувачі освіти, які приймають участь у конференціях та мають відповідні тези доповідей та публікації, що співвідносяться з тематикою курсу, за погодженням з викладачем можуть отримати додатково 5 балів до результатів оцінювання за поточним контролем.

7.6. Визнання результатів навчання, отриманих у неформальній освіті. За результатами участі у заходах неформальної освіти: вебінарах, тренінгах, курсах і т.ін., при наявності підтверджуючих документів (сертифікати), здобувачам освіти можуть бути зараховані певні складові освітньої компоненти, що співвідносяться із тематикою.

8. Рекомендовані джерела інформації

Базові:

1. Андрій Будай. Дизайн-патерни — просто, як двері [Електронний ресурс]. – Режим доступу: <https://programming.in.ua/programming/basisprogramming/268-design-patterns-book-andriy-buday.html>
2. Карпенко М. Ю. Технології створення програмних продуктів та

- інформаційних систем : навч. посібник / М. Ю. Карпенко, Н. О. Манакова, І. О. Гавриленко ; Харків. нац. ун-т міськ. госп-ва ім. О. М. Бекетова. – Харків: ХНУМГ ім. О. М. Бекетова, 2017. – 93 с.
3. Кузьмних В. О. Моделі та засоби управління ІТ-проєктами» [Текст]: навч.посіб./ В. О. Кузьмних, О. В. Коваль, Р. А. Тараненко. – К.: КПІ ім. Ігоря Сікорського, 2023. – 222 с
 4. Рефакторинг // Refactoring.Guru [Electronic resource]. – Access mode: URL: <https://refactoring.guru/uk/refactoring>
 5. Патерни проектування [Електронний ресурс]. – Режим доступу: <https://abitap.com/category/paterny-proektuvannya/>
 6. Refactoring. Improving the Design of Existing Code Second Edition Martin Fowler with contributions by Kent Beck [Electronic resource]. – Access mode: <https://dl.ebooksworld.ir/motoman/Refactoring.Improving.the.Design.of.Existing.Code.2nd.edition.www.EBooksWorld.ir.pdf>
 7. Robert C. Martin. Clean code. Creating and refactoring with Agile [Electronic resource]. – Access mode: https://github.com/martinmurciego/good-books/blob/master/Clean%20Code_%20A%20Handbook%20of%20Agile%20Software%20Craftsmanship%20-%20Robert%20C.%20Martin.pdf
 8. Clausen Ch. Five Lines of Code: How and when to refactor // Manning, 2021 — 338 p. — ISBN 9781617298318
 9. Андрій Будай. Дизайн-патерни — просто, як двері [Електронний ресурс].— Режим доступу: <https://programming.in.ua/programming/basisprogramming/268-design-patterns-book-andriy-buday.html>
 10. Claude Y. Laporte Alain Software Quality Assurance, First Edition/ USA: Wiley-IEEE Press, 2017. – 596 p.
 11. Daniel Galin Software Quality Concepts and Practice / USA.: Wiley-IEEE Press, 2018. – 711 p.
 12. Stefan Wolpers. The Scrum Guide 2020 – Reordered [Electronic resource].— Access mode:<https://www.mateuszeromski.pl/wp-content/uploads/2021/04/Agile-of-Product-com-Scrum-Guide-2020-Reordered-v103.pdf>
 13. Eric Enge, Stephan Spencer, and Jessie C. Stricchiola SEO: The Art of SEO: Mastering Search Engine Optimization Third Edition. O`Reilly Media, Inc., 2017. 162p.

Додаткові:

1. Спирінцев В.В. Управління ІТ-проєктами [Текст]: навч.посіб./ В.В.Спирінцев, В.В., Гнатушенко, Д.В.Суцевський, Т.А. Прокоф'єв.-Д.:РВВ ДНУ, 2018.-112с.
2. Спирінцев В.В. Web-технології та Web-дизайн: HTML, CSS [Текст]: навч.посіб./В.В.Спирінцев, В.В.Гнатушенко, О.С.Волковський.-Дніпро: 2017.-163с.
3. Авраменко А.С., Авраменко В.С., Косенюк Г.В. Тестування програмного забезпечення. Навчальний посібник/ А.С.Авраменко, В.С.Авраменко, Г.В. Косенюк. – Черкаси: ЧНУ імені Богдана Хмельницького, 2017. – 284 с.

4. Svyatoslav Kulikov. Software testing. Base course, 3rd Edition/ EPAM Systems, 2022.-278p.
5. Selenium documentation [Electronic resource]. – Access mode: <https://www.selenium.dev/documentation/>-Title from the screen.
6. Scott Chacon, Ben Straub. Pro Git, Second Edition/Apress, 2014.- 495p.