

	Силабус навчальної дисципліни <u>«Розробка клієнт-серверних застосувань»</u>
Статус дисципліни	Навчальна дисципліна є <i>нормативною</i>
Курс	3
Семестр	5-6
Обсяг дисципліни, кредити ЄКТС / загальна кількість годин	6 кредитів /180 год.
Мова викладання	українська
Що буде вивчатися (предмет навчання)	<p>1.1 Предметом вивчення навчальної дисципліни "Розробка клієнт-серверних застосувань" є вивчення технологій, методів та практичних навичок, пов'язаних із проектуванням, розробкою та впровадженням програм, що базуються на клієнт-серверній архітектурі. У рамках цієї дисципліни студенти ознайомлюються з основними принципами взаємодії між клієнтами та серверами через мережу, вчать розробляти клієнтські та серверні компоненти, забезпечувати безпеку, працювати з базами даних, а також реалізовувати різноманітні клієнт-серверні застосунки, що використовуються в сучасних інформаційних системах.</p> <p>1.2. Основними завданнями вивчення дисципліни «Розробка клієнт-серверних застосувань» є надання студентам поглиблених знань та навичок для розуміння, проектування, розробки та впровадження програм, які базуються на клієнт-серверній архітектурі. Ця дисципліна спрямована на підготовку студентів до успішної роботи з високорівневими технологіями та методами, які застосовуються у реальних клієнт-серверних проектах.</p>
Чому це цікаво / необхідно вивчити (мета) доступом	<p>Метою даної дисципліни є формування в студентів глибокого розуміння концепцій, технологій та методів, пов'язаних з проектуванням, розробкою та впровадженням клієнт-серверних застосунків. Студенти мають набути практичних навичок у створенні робочих клієнтських та серверних компонентів, здатних ефективно взаємодіяти через мережу.</p> <p>Вивчення дисципліни "Розробка клієнт-серверних застосувань" цікаве і необхідне, оскільки вона надає студентам важливі знання та навички, що стосуються сучасної розробки програмного забезпечення. Спроможність проектувати, розробляти та впроваджувати клієнт-серверні застосунки є важливою у сучасному інформаційному світі, де такі застосунки широко використовуються у веб-розробці, мобільних додатках, інтернет-послугах та багатьох інших галузях. Навички, набуті в ході вивчення цієї дисципліни, дозволять студентам реалізовувати інноваційні інформаційні проекти та розробляти рішення, які забезпечують ефективну комунікацію між клієнтами та серверами, що має велике практичне застосування в сучасному світі програмування.</p>

<p>Чому можна навчитись (компетентності)</p>	<p>Інтегральна компетентність</p> <p>Здатність вирішувати типові спеціалізовані задачі в галузі інформаційних технологій або у процесі навчання, що вимагає застосування положень і методів комп'ютерних наук та може характеризуватися певною невизначеністю умов; нести відповідальність за результати своєї діяльності; здійснювати контроль інших осіб у визначених ситуаціях.</p> <p>Загальні компетентності:</p> <p>ЗК5. Знання та розуміння предметної області та розуміння професійної діяльності.</p> <p>ЗК8. Здатність вчитися і оволодівати сучасними знаннями;</p> <p>СК2. Здатність використовувати теоретичні та фундаментальні знання в галузі комп'ютерних наук та інформаційних технологій для вирішення різноманітних проблем.</p> <p>СК3. Здатність розробляти, аналізувати та застосовувати ефективні алгоритми для розв'язання конкретних професійних задач залежно від предметного середовища.</p> <p>СК4. Здатність здійснювати проектування та розробку програмного забезпечення</p> <p>СК8. Здатність застосовувати сучасні методи, технології та інструментальні засоби проектування й створення програмних систем та їх супроводження</p> <p>СК9. Здатність застосовувати знання сучасних методів і технологій створення та супроводження розподілених систем.</p> <p>СК10. Здатність адмініструвати системне та прикладне програмне забезпечення під час реалізації процесів життєвого циклу інформаційних систем</p> <p>Результати навчання:</p> <p>РН05. Розуміти основні методи і технології об'єктно-орієнтованого та компонентного програмування.</p> <p>РН08. Розробляти застосунки, використовуючи сучасні веб-технології.</p> <p>РН10. Знати методології, методи, моделі, процеси і технології життєвого циклу розробки та тестування програмного забезпечення.</p> <p>РН11. Застосовувати сучасні мови програмування та технології для розробки програмного забезпечення розподілених систем.</p> <p>РН12. Знати основні принципи функціонування системного та прикладного програмного забезпечення.</p>
<p>Як можна користуватись набутими знаннями і вміннями (результати навчання)</p>	<p>Засвоєння освітньої компоненти впливає на такі результати навчання:</p> <p>Професійні навички: Засвоєння освітньої компоненти впливає на професійні навички студентів, дозволяючи їм набути глибоке розуміння та практичну експертизу у проектуванні, розробці та впровадженні клієнт-серверних програм. Вони опановують навички роботи з різними мовами програмування, фреймворками та технологіями, забезпечуючи ефективну взаємодію між клієнтами та серверами, враховуючи аспекти безпеки, роботи з базами даних і створення веб-додатків. Ці навички роблять студентів більш конкурентоспроможними на ринку праці та допомагають їм реалізувати інноваційні інформаційні проекти в різних галузях, де важлива розробка клієнт-серверних рішень.</p> <p>Ефективність роботи: Засвоєння освітньої компоненти впливає на ефективність роботи студентів, оскільки вони набувають навичок та знань, необхідних для проектування та розробки клієнт-серверних рішень. Ці навички дозволяють їм створювати програми та застосунки, які ефективно взаємодіють через мережу, оптимізують обмін даними та</p>

	ресурсами між клієнтами та серверами. В результаті, студенти стають здатними розробляти рішення, які працюють ефективно, швидко та надійно, що є важливим фактором в сучасному інформаційному середовищі, де швидкість та надійність взаємодії між програмами та користувачами мають вирішальне значення для успіху проектів і підприємств.
Пререквізити	«Алгоритмізація та програмування».
Постреквізити	«Безпека розробки та підтримки додатків»/«Основи і стандарти інформаційної безпеки»/«Захист інформації веб додатку», «Технологія захисту інформації», «Переддипломна практика», «Кваліфікаційна робота»
Навчальна логістика	<p>Тема 1. Вступ до клієнт-серверної архітектури.</p> <p>Тема 2. Вступ до мови Java.</p> <p>Тема 3. Змінні, типи даних та операції.</p> <p>Тема 4. Умови та цикли.</p> <p>Тема 5. Функції та процедури.</p> <p>Тема 6. Масиви.</p> <p>Тема 7. Сортування масивів.</p> <p>Тема 8. Принципи ООП.</p> <p>Тема 9. Класи та об'єкти</p> <p>Тема 10. Наслідування</p> <p>Тема 11. Поліморфізм</p> <p>Тема 12. Абстрактні класи</p> <p>Тема 13. Інтерфейси.</p> <p>Тема 14. Огляд ієрархії колекцій.</p> <p>Тема 15. ArrayList.</p> <p>Тема 16. LinkedList.</p> <p>Тема 17. HashSet.</p> <p>Тема 18. HashMap</p> <p>Тема 19. Популярні паттерни для клієнт-серверної архітектури.</p> <p>Тема 20. Технології інтеграції з базами даних.</p> <p>Тема 21. RESTful API.</p>

<p>Інформаційне забезпечення</p>	<p>3. Рекомендована література</p> <p>Базова</p> <ol style="list-style-type: none"> 1. Bloch J. Effective Java: 3rd Edition, Addison Wesley, 2017, 412 p. 2. Schildt H. Java: A Beginner's Guide: 8th Edition, McGraw-Hill Education, 2018, 684 p. 3. Schildt H. Java: The Complete Reference: 11th Edition, McGraw-Hill Education, 2018, 1208 p. 4. Horstmann C. S. Core Java Volume I – Fundamentals: 11th Edition, Prentice Hall 2018, 889 p. 5. Horstmann C. S. Core Java SE 9 for the Impatient: 2nd Edition Addison-Wesley Professional, 2017, 576 p. 6. Eckel B. Thinking in Java 4th Edition: Pearson, 2006, 1150 p. 7. Deitel P., Deitel H. Java How to Program, Early Objects: 11th Edition, Pearson, 2017, 1296 p. 8. Deitel P., Deitel H. Java How To Program, Late Objects: 11th Edition, Pearson, 2017, 1248 p. 9. Ратушняк Т. В. Програмування мовою JAVA: практикум: навчальний посібник. Державна фіскальна служба України, Університет державної фіскальної служби України. Ірпінь, 2017. 212 с. 10. Копитко М.Ф., Іванків К.С. Основи програмування мовою Java: Тексти лекцій. – Львів: Видавничий центр ЛНУ ім. Івана Франка, 2002. – 83 с. 11. Брнакевич І.Є., Вагін П.П. Програмування мовою Java: використання фундаментальних класів: Тексти лекцій. – Львів: Видавничий центр ЛНУ імені Івана Франка, 2002. – 75 с. <p>Додаткова</p> <ol style="list-style-type: none"> 1. Освоюємо Java – Вікіпідручник // http://uk.wikibooks.org/wiki/Освоюємо_Java 2. Програмування на Java // http://javaland.com.ua 3. Брнакевич І.Є., Вагін П.П. Програмування мовою Java: використання фундаментальних класів: Тексти лекцій // http://blues.franko.lviv.ua/ami/books/ami/Java_fundamental.pdf 4. Java Tutorials // http://docs.oracle.com/javase/tutorial (англ.) 5. Java Tutorial // https://www.w3schools.com/java/ (англ.) 6. Java Tutorial // http://www.java2s.com/Tutorial/Java/CatalogJava.htm (англ.) 7. Learn Java Programming//https://www.programiz.com/java-programming (англ.) 8. Java Programming Language // https://www.geeksforgeeks.org/java/ (англ.) 9. Bruce Eckel. On Java 8 // http://www.sufuq.com/books/java/On%20Java%208.pdf.
---	---

<p>Політика навчальної дисципліни, оцінювання результатів навчання та академічна доброчесність</p>	<p><i>Політика щодо відвідування та проведення занять.</i> Під час лекцій, практичних та лабораторних занять використовуються різноманітні інтерактивні технології навчання, які допомагають не тільки засвоїти теми курсу, а й розвинути навички критичного мислення, уміння працювати з інформацією, презентувати результати власних досліджень.</p> <p>Передбачається обов'язкова присутність студента на кожному занятті, тому що для отримання ефекту занурення у проблематику дисципліни необхідне групове обговорення певних завдань та шляхи їх вирішення («мозковий штурм»).</p> <p>Слід відзначити, що через відсутність студента на занятті можна втратити логіку опанування теоретичного та практичного матеріалу, якою пов'язані всі теми курсу. Як правило, викладач попереджає це на вступній лекції, на якій відбувається знайомство зі структурно-логічною схемою курсу.</p> <p>У випадку, якщо була поважна причина відсутності студента на занятті, необхідно відвідати консультацію та з викладачем обговорити проблемні питання теми або низки тем через розбір «скрізних» питань, виконати практичні завдання.</p> <p>Під час вивчення курсу можна використовувати як рекомендовану літературу, так й різні інформаційні ресурси. Викладач контролює якість інформації, яку використовують здобувачі під час виконання завдань, вчить їх працювати з науковою інформацією, формує навички відрізняти якісну інформацію від неякісної. Мобільні пристрої під час проведення занять дозволяється використовувати лише для навчальних та наукових цілей.</p> <p><i>Політика щодо академічної доброчесності.</i> Політика щодо академічної доброчесності побудована на основі Положення про академічну доброчесність в ВСП «ФКЗІ ДУІТЗ». Усі види письмових робіт повинні бути написані здобувачами самостійно та мати високий рівень оригінальності.</p> <p>До заліку допускаються здобувачі, які виконали індивідуальне завдання. Здобувач, який не з'явився на залік або не був допущений на момент його проведення, має право повторно його пройти у визначені викладачем терміни під час консультацій/ відпрацювань.</p> <p><i>Загальна оцінка з дисципліни</i> – максимум 100 балів. У випадку отримання менше 60 балів, здобувач обов'язково здійснює перескладання для ліквідації академічної заборгованості.</p>
<p>Локація та</p>	<p><i>Навчальна аудиторія</i> (дошка, проектор, ноутбук, інше)</p>

матеріально-технічне забезпечення	обладнання). <i>Дистанційна</i> – сучасні платформи та онлайн-сервіси дистанційного навчання.	
Семестровий контроль	<i>екзамен</i>	
Циклова комісія	інформаційних технологій	
Викладач		ПІБ Сорока Ігор Сергійович
		викладач
		E-mail: ishtvar@gmail.com