

Силабус курсу

Міждисциплінарна курсова робота



Освітньо-професійна програма – «Інженерія програмного забезпечення»

Ступінь вищої освіти – магістр

Спеціальність – 121 Інженерія програмного забезпечення

Рік навчання: 1, Семестр: 1

Кількість кредитів: 3

Мова викладання: українська

Керівник курсу

Кафедра комп'ютерних наук

Контактна інформація

kn@wunu.edu.ua, (0352) 23-60-29

Опис дисципліни

Міждисциплінарна курсова робота є одним з видів наукової роботи. Самостійним навчально-науковим дослідженням студента, виконується на першому курсі навчання та поєднує декілька дисциплін професійної підготовки магістра.

Метою виконання міждисциплінарної курсової роботи є систематизація, закріплення та розширення теоретичних знань, їхнє застосування для вирішення конкретного практичного завдання відповідно до вимог ОПП зі спеціальності 121 Інженерія програмного забезпечення.

Міждисциплінарна курсова робота дає змогу виявити здатність студента самостійно осмислити проблему, творчо, критично її дослідити, набути вміння збирати, аналізувати і систематизувати інформацію та літературні джерела, застосовувати отримані знання при вирішенні практичних завдань, а також формує наступні фахові компетентності:

- здатність аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення;
- здатність розробляти і реалізовувати наукові та/або прикладні проекти у сфері інженерії програмного забезпечення;
- здатність проектувати архітектуру програмного забезпечення, моделювати процеси функціонування окремих підсистем і модулів;
- здатність розвивати і реалізовувати нові конкурентоспроможні ідеї в інженерії програмного забезпечення;
- здатність розробляти, аналізувати та застосовувати специфікації, стандарти, правила і рекомендації в сфері інженерії програмного забезпечення;
- здатність критично осмислювати проблеми у галузі інформаційних технологій та на межі галузей знань, інтегрувати відповідні знання та розв'язувати складні задачі у широких або мультидисциплінарних контекстах;
- здатність розробляти і координувати процеси, етапи та ітерації життєвого циклу програмного забезпечення на основі застосування сучасних моделей, методів та технологій розроблення програмного забезпечення;
- здатність забезпечувати якість програмного забезпечення;
- здатність розробляти програмне забезпечення, використовуючи концепції інформаційної безпеки;
- здатність використовувати технології що забезпечують цілісність баз даних та мережеву безпеку;
- здатність узагальнювати результати наукової та проектної діяльності;
- здатність використовувати новітні підходи у сфері інформаційних технологій для опрацювання даних, отриманих в умовах інтервальної невизначеності.

Структура курсу

Виконання міждисциплінарної курсової роботи передбачає такі етапи:

- 1) вибір та затвердження теми міждисциплінарної курсової роботи;
- 2) аналіз сучасних професійних стандартів і інших нормативно-правових документів з інженерії програмного забезпечення;
- 3) складання плану роботи;
- 4) написання та оформлення тексту міждисциплінарної курсової роботи;
- 5) подання завершеної міждисциплінарної курсової роботи на кафедру для рецензування та перевірки дотримання студентом необхідної кількості авторського тексту;
- 6) доопрацювання міждисциплінарної курсової роботи (у разі необхідності) згідно із зауваженнями керівника;
- 7) захист міждисциплінарної курсової роботи.

Пререквізити

Успішне вивчення дисциплін, передбачених навчальним планом ОПП зі спеціальності 121 Інженерія програмного забезпечення.

Постреквізити

Можливість здійснення подальшої науково-дослідної роботи за спеціальністю.

Результати навчання

В результаті виконання міждисциплінарної курсової роботи здобувачі освітньо-професійної програми «Інженерія програмного забезпечення» мають вміння:

- знати і застосовувати сучасні професійні стандарти і інші нормативно-правові документи з інженерії програмного забезпечення;
- оцінювати і вибирати ефективні методи і моделі розроблення, впровадження, супроводу програмного забезпечення та управління відповідними процесами на всіх етапах життєвого циклу;
- будувати і досліджувати моделі інформаційних процесів у прикладній області;
- виявляти інформаційні потреби і класифікувати дані для проектування програмного забезпечення;
- розробляти, аналізувати, обґрунтовувати та систематизувати вимоги до програмного забезпечення;
- розробляти і оцінювати стратегії проектування програмних засобів; обґрунтовувати, аналізувати і оцінювати варіанти проектних рішень з точки зору якості кінцевого програмного продукту, ресурсних обмежень та інших факторів;
- аналізувати, оцінювати і застосовувати на системному рівні сучасні програмні та апаратні платформи для розв'язання складних задач інженерії програмного забезпечення;
- розробляти і модифікувати архітектуру програмного забезпечення для реалізації вимог замовника;
- обґрунтовано вибирати парадигми і мови програмування для розроблення програмного забезпечення; застосовувати на практиці сучасні засоби розроблення програмного забезпечення.
- модифікувати існуючі та розробляти нові алгоритмічні рішення детального проектування програмного забезпечення;
- забезпечувати якість на всіх стадіях життєвого циклу програмного забезпечення, у тому числі з використанням релевантних моделей та методів оцінювання, а також засобів автоматизованого тестування і верифікації програмного забезпечення;
- приймати ефективні організаційно-управлінські рішення в умовах невизначеності та зміни вимог, порівнювати альтернативи, оцінювати ризики;
- конфігурувати програмне забезпечення, керувати його змінами та розробленням програмної документації на всіх етапах життєвого циклу;
- прогнозувати розвиток програмних систем та інформаційних технологій;
- здійснювати реінжиніринг програмного забезпечення відповідно до вимог замовника;
- планувати, організувати та здійснювати тестування, верифікацію та валідацію програмного забезпечення;
- збирати, аналізувати, оцінювати необхідну для розв'язання наукових і прикладних задач інформацію, використовуючи науково-технічну літературу, бази даних та інші джерела;
- планувати, організувати, впроваджувати та контролювати розробку програмного забезпечення систем захисту інформації. рн19. використовувати концепції інформаційної безпеки, безпеки баз даних, мережевої безпеки та криптографії;
- робити висновки з результатів наукової та проектної діяльності, готувати наукові публікації, представляти результати досліджень;
- використовувати новітні підходи у сфері інформаційних технологій для опрацювання даних, отриманих в умовах інтервальної невизначеності.

Інформація про дисципліну

Ступінь вищої освіти	Магістр
Спеціальність	121 Інженерія програмного забезпечення
Галузь знань	12 Інформаційні технології
Курс (рік навчання)	1
Семестр	1
Рік викладання	2021-2022
Форма навчання	Денна
Нормативна / вибіркова	Нормативна
Загальна кількість год. / кредитів	90 / 3

Літературні джерела

1. Чистий код. Створення і рефакторинг за допомогою Agile / Роберт Сесіл Мартін. – «Фабула», 2019. – 448 с.
2. Рефакторинг. Поліпшення існуючого коду / М. Фаулер, К. Бек, Дж. Брант, В. Опдайк, Д. Робертс. – «Діалектика», 2003. – 448 с.
3. Авраменко О.А. Архітектура засобів редокументування успадкованого програмного забезпечення. — Вісн. НАУ. — 2007.— № 3-4 (33). — С. 58—62.
4. Michael Feathers / Working Effectively with Legacy Code. - Person, 2004. – 464с.
5. 10 Idiomatic Ways to Refactor Your Python Code / Yong Cui. – [онлайн ресурс] - <https://towardsdatascience.com/10-idiomatic-ways-to-refactor-your-python-code-cbb05bb0c820> - 3.07.2000р.
6. Програмна інженерія: підруч. / К.М. Лавріщева; Ін-т програм. систем НАН України. — К.: Академперіодика, 2008. — 320 с.
7. Craig Larman Applying UML and Patterns. An Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition), 2004, 736 с.
8. Rumbaugh J., Jacobson I., Booch G. The Unified Modeling Language Reference Manual, 2nd Ed. — Boston: Addison-Wesley, 2004. — 742 p.
9. G. Hohpe, B. Woolf. Enterprise Integration Patterns : Designing, Building, and Deploying Messaging Solutions. Addison-Wesley, 2011. 688с.
10. Guckenheimer S., Perez J.I. Software Engineering with Microsoft Studio Team System. Crawfordsville:Adison–Wesley, 2006. — 304 p.
11. Microsoft Security Development Lifecycle (SDL) – Process Guidance <https://msdn.microsoft.com/en-us/library/windows/desktop/84aed186-1d75-4366-8e61-8d258746bopq.aspx>
12. OWASP Foundation. OWASP Testing Guide v4.0. URL: https://www.owasp.org/index.php/Web_Application_Penetration_Testing.
13. J. Koo, Y. Kim and S. Lee, "Security Requirements for Cloud-based C4I Security Architecture", 2019 International Conference on Platform Technology and Service (PlatCon), pp. 1-4, 2019.
14. C. Bryce, "Security governance as a service on the cloud", J Cloud Comp, vol. 8, 2019.
15. G. Levitin, L. Xing and H.Z. Huang, "Security of separated data in cloud systems with competing attack detection and data theft processes", Risk Analysis, vol. 39, no. 4, pp. 846-858, 2019.
16. K. O'Loughlin, M. Neary, E.C. Adkins and S.M. Schueller, "Reviewing the data security and privacy policies of mobile apps for depression", Internet interventions, vol. 15, pp. 110-115, 2019.
17. B Ouyang and Y. Cui, "Research on Computer Network Security Prevention in the Era of Big Data[J]", Journal of Physics: Conference Series, vol. 1648, no. 2, pp. 022011, 2020.
18. C. Wang, S. Chen, Z. Feng, Y. Jiang and X. Xue, "Block Chain-Based Data Audit and Access Control Mechanism in Service Collaboration", 2019 IEEE International Conference on Web Services, pp. 214-218, 2019.
19. Shevchuk R. Software for Automatic Estimating Security Settings of Social Media Accounts / R. Shevchuk, A. Melnyk, O. Opalko, H. Shevchuk // Proceedings of the 2020 10th International Conference “Advanced Computer Information Technologies” – Deggendorf, Germany. – September 16–18, 2020 – P. 769 – 773.
20. Cheshun V. Safe Decentralized Applications Development Using Blockchain Technologies / V. Cheshun, I. Muliar, V. Yatskiv, R. Shevchuk, S. Kulyna // Proceedings of the 2020 10th International Conference “Advanced Computer Information Technologies” – Deggendorf, Germany. – September 16–18, 2020 – P.800 – 805.
21. Wojtowicz M. Monte Carlo Type Method of Attack on the RSA Cryptosystem / M. Wojtowicz, D. Bodnar, R. Shevchuk, O. Bodnar, I. Bilanyk // // Proceedings of the 2020 10th International Conference “Advanced Computer Information Technologies” – Deggendorf, Germany. – September 16–18, 2020 – P.755 – 758.
22. Shevchuk R. Improve the Security of Social Media Accounts / R. Shevchuk, Y. Pastukh // Proceedings of the 2019 9th International Conference “Advanced Computer Information Technologies” – Ceske Budejovice, Czech Republic. – June 5–7, 2019 –P.439-442.

Система оцінювання та вимоги

Перелік документів, необхідних для виконання та захисту міждисциплінарної курсової роботи поданий у методичних вказівках до виконання міждисциплінарної курсової роботи. Роботи, переписані з літературних джерел, нормативних документів, неопрацьовані і неоформлені належним чином, виконані шляхом запозичення ідей інших авторів без посилання на використані джерела, до захисту не допускаються.

Оцінювання

Остаточна оцінка за курс розраховується наступним чином:

Види оцінювання	% від остаточної оцінки
Підготовка тексту роботи	40
Захист роботи	60

Шкала оцінювання студентів:

ECTS	Бали	Зміст
A	90-100	відмінно
B	85-89	добре
C	75-84	добре
D	65-74	задовільно
E	60-64	достатньо
FX	35-59	незадовільно з можливістю повторного складання
F	1-34	незадовільно з обов'язковим повторним курсом