



Силабус курсу

ТЕХНОЛОГІЇ ПРОГРАМУВАННЯ ДЛЯ КОМП'ЮТЕРНИХ СИСТЕМ ТА МЕРЕЖ

Освітньо-професійна програма «Комп'ютерна інженерія»

Ступінь вищої освіти – бакалавр

Спеціальність: 123 «Комп'ютерна інженерія»

Рік навчання: 4, Семестр: 2

Кількість кредитів: 5, Мова викладання: українська

Керівник курсу

ПП

к.т.н., доцент Юрій Батько

Контактна інформація bum@wunu.edu.ua

Опис дисципліни

Дисципліна „Технології програмування для комп'ютерних систем та мереж” орієнтовані на глибоке та ґрунтовне засвоєння студентами систематичних знань та практичних навичок використання сучасних програмно-апаратних засобів для проектування, реалізації, верифікації, оптимізації та опису програмних систем з подальшим використанням набутих навичок під час розв'язання актуальних завдань. Завдання курсу полягає в вивченні науково-практичного інструментарію проектування прикладних програмних систем та їх складових за допомогою сучасних засобів автоматизованого проектування.

Структура курсу

№ п/п	Тема	Результати навчання	Завдання
1.	Основні поняття програмної інженерії	Засвоїти основні поняття та задачі курсу, особливості проектування складних програмних систем	Питання
2.	Процес створення ПЗ.	Засвоїти існуючі підходи та технології до створення програмних продуктів	Питання, лабораторна робота
3.	Міжнародні стандарти життєвого циклу ПЗ	Ознайомитись існуючими моделями проектування та створення складних програмних продуктів	Питання
4.	Принципи та методології реалізації ПЗ	Вивчити існуючих технології програмної реалізації складних програмних системи (швидка розробка, екстремальне проектування тощо)	Питання
5.	Патерни програмування	Вивчити існуючі патерни для проектування та реалізації програмних	Питання, лабораторна

	та приклади реалізації	систем	робота
6.	Інженерія вимог до ПЗ	Ознайомитись з принципами та механізмами та специфікаціями вимог до програмних розробок	Питання, лабораторна робота
7.	Проектування архітектури ПЗ.	Вивчити технології та принципи проектування й реалізації внутрішньої архітектури програмних додатків	Питання, лабораторна робота
8.	Проектування інтерфейсу користувача.	Вивчити технології та принципи проектування й реалізації графічних інтерфейсів користувача програмних додатків	Питання, лабораторна робота
9.	Класичні методи аналізу і проектування ПЗ.	Вивчити методи для проведення моделювання внутрішньої архітектури складних програмних систем	Питання, лабораторна робота
10.	Об'єктно-орієнтований аналіз і проектування.	Освоїти принципи та механізми використання об'єктно-орієнтованого підходу при проектування складних програмних систем	Питання, лабораторна робота
11.	Архітектура розподілених програмних систем.	Вивчити особливості проектування та реалізації програмних систем з елементами розподілення обробки даних	Питання
12.	Управління програмними проектами.	Вивчити механізми та сучасні системи автоматизації управління програмними проектами	Питання, лабораторна робота
13.	Управління ризиками в програмних проектах.	Вивчити механізми та сучасні системи автоматизації управління ризиками в програмних проектах	Питання
14.	Системи управління дефектами	Вивчити механізми та сучасні системи автоматизації управління дефектами в програмних проектах	Питання
15.	Контроль версій в програмних проектах	Вивчити механізми та сучасні технології автоматизації контролю версій в програмних проектах	Питання, лабораторна робота
16.	Збір і випуск ПЗ.	Вивчити механізми та сучасні технології автоматизації під час збору програмних проектів	Питання, лабораторна робота
17.	Якість програмного забезпечення	Засвоїти алгоритми та програмні засоби для проведення тестування та оцінки якості окремих модулів програмних систем	Питання, лабораторна робота
18.	Інструментальні засоби інженерії програмного забезпечення.	Ознайомитись з сучасними інструментальними засобами інженерії програмного забезпечення, їхніми технічними можливостями та особливостями	Питання, лабораторна робота

Літературні джерела

1. Alexander Stepanov and Paul McJones Elements of Programming Semigroup Press, 2019, 218
2. Kenneth Leroy Busbee, Dave Braunschweig. Programming Fundamentals - A Modular Structured Approach, 2nd Edition Houston Community College; eBook (Creative Commons Licensed)
3. Johan Sannemo Principles of Algorithmic Problem Solving KTH Royal Institute of Technology, 2018, 351.
4. Divakar Viswanath Scientific Programming and Computer Architecture. The MIT Press, 2017, 434.
5. Frank B. Brokken C++ Annotations: An Extensive Tutorial about the C++ Programming Language. University of Groningen, 2021.
6. Bjarne Stroustrup, Herb Sutter C++ Core Guidelines Standard C++ Foundation and GitHub. 2022.
7. Matthias Felleisen, Robert Bruce Findler, Matthew Flatt, Shriram Krishnamurthi How to Design Programs, second edition: An Introduction to Programming and Computing (The MIT Press) second edition 2018, 792.
8. Richard Jones, Antony Hosking, Eliot Moss. The Garbage Collection Handbook ("International Perspectives on Science, Culture and Society") 1st Edition, Chapman and Hall/CRC; 1st edition (September 30, 2020), 2020, 520.
9. Vaskaran Sarcar Java Design Patterns: A Hands-On Experience with Real-World Examples 2nd ed. Edition, Apress; 2nd ed. edition (December 7, 2018), 2018, 533.
10. Mykel Kochenderfer, Tim Wheeler, and Kyle Wray. Algorithms for Decision Making, The MIT Press (August 2, 2022); eBook (Creative Commons Edition). 2022. 704.
11. James Aspnes Notes on Data Structures and Programming Techniques, Yale University (Creative Commons Attribution-ShareAlike 4.0 (CC BY-SA 4.0)) (2022), 2022. 528 с.
12. Matthias Felleisen, Robert Bruce Findler, Matthew Flatt, Shriram Krishnamurthi How to Design Programs, second edition: An Introduction to Programming and Computing (The MIT Press) second edition 2018, 792.
13. Florian Jatton The Constitution of Algorithms: Ground-Truthing, Programming, Formulating. The MIT Press, 2021, 154.
14. Xinyu Liu. Elementary Algorithms. GitHub.com, 2021, 503.
15. Granville Barnett, Luca Del Tongo, John Bullinaria. Data Structures and Algorithms: Annotated Reference with Examples. University of Birmingham, 2019, 112.

Політика оцінювання

Політика щодо дедлайнів та перескладання: Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку (-20 балів). Перескладання модулів відбувається із дозволу деканату за наявності поважних причин (наприклад, лікарняний).

Політика щодо академічної доброчесності: Списування під час залікових модулів та екзаменів заборонені (в т.ч. із використанням мобільних девайсів).

Політика щодо відвідування: Відвідування занять є обов'язковим компонентом оцінювання, за яке нараховуються бали. За об'єктивних причин (наприклад, хвороба, міжнародне стажування) навчання може відбуватись в он-лайн формі за погодженням із керівником курсу.

Оцінювання

Заліковий модуль 1	Заліковий модуль 2	Заліковий модуль 3 (підсумкова оцінка за КПІЗ)	Заліковий модуль 4 (екзамен)	Разом
20 %	20 %	20 %	40 %	100%
1. Усне опитування під час заняття (8 тем по 5 бали = 40 балів) 2. Письмова робота = 60 балів	1. Усне опитування під час заняття (10 тем по 5 бали = 50 балів) 2. Письмова робота = 50 балів	1. Написання та захист КПІЗ = 80 балів. 2. Виконання завдань під час тренінгу = 20 балів	1. Тестові завдання (25 тестів по 2 бали за тест) – макс. 50 балів 2. Завдання. 1 – макс. 25 балів 3. Завдання. 2 – макс. 25 балів	100

За шкалою університету ЗУНУ	За національною шкалою	За шкалою ECTS
90-100	Відмінно	A (відмінно)
85-89	Добре	B (дуже добре)
75-84		C (добре)
65-74	Задовільно	D (задовільно)
60-64		E (достатньо)
35-59	Незадовільно	FX (незадовільно, з можливістю повторного складання)
1-34		F (незадовільно, з обов'язковим повторним курсом)