



Силабус курсу

Технологія розробки програмного забезпечення

Ступінь вищої освіти - магістр

Спеціальність: 122 «Комп'ютерні науки»

Освітньо-професійна програма: «Комп'ютерні науки»

Рік навчання: 3 Семестр: 6

Кредитів: 8 Мова викладання: українська

Керівник курсу

ПШ

викладач, Віталій ДОРОШ

Контактна інформація

vdo@wunu.edu.ua

Опис дисципліни

"Технологія розробки програмного забезпечення" - це дисципліна, що вивчає процес розробки програмного забезпечення від постановки завдання до його випуску та підтримки. Вона включає в себе методи та інструменти, що використовуються в програмному проектуванні та розробці.

Дисципліна "Технологія розробки програмного забезпечення" має на меті ознайомити студентів з сучасними методами та технологіями розробки програмного забезпечення, включаючи архітектуру програмного забезпечення, процес розробки, методології, безпеку, роботу в команді та розгортання програмного забезпечення. Дисципліна спрямована на формування знань та практичних навичок, які можуть бути застосовані у роботі з розробкою програмного забезпечення.

Структура курсу

Години (лек./лаб.)	Тема	Результати навчання	Завдання
2/-	Тема 1. Базові поняття створення програмних продуктів	Знати поняття моделі життєвого циклу програмного забезпечення та інформаційної системи, основні етапи життєвого циклу інформаційної системи – аналіз, розроблення вимог, проектування, тестування, ввід у експлуатацію, супровід. Знати причини виникнення технології програмної інженерії, структуру життєвого циклу за міжнародним стандартом ISO/IEC 12207. Знати концепцію інформаційної безпеки та принципи безпечного проектування програмного забезпечення.	Питання
2/6	Тема 2. Класифікація та сутність процесів розроблення програмного забезпечення	Знати принципи і методи проектування програмного забезпечення, методи програмної інженерії, рівні управління вимогами та особливості застосування та формулювання вимог.	Питання Практична робота
4/-	Тема 3. Моделі життєвого циклу ПЗ.	Знати технології розроблення програмного забезпечення та програмний продукт. Вміти застосовувати життєві цикли програмного	Питання

		забезпечення, моделі життєвого циклу. Знати стратегії розробки програмного забезпечення.	
2/7	Тема 4. Гнучкі методології.	Знати методології Scrum, Agile, Kanban. XP Scrum.	Питання Практична робота
2/-	Тема 5. Інженерія вимог до ПЗ.	Знати методики формування вимог, вимоги до програмного забезпечення, категорії вимог, етапи інженерії вимог, типи вимог, формування вимог, аналіз вимог (класичний підхід). Вміти робити опис вимог у вигляді текстових сценаріїв, формування та аналіз вимог в процесі Scrum.	Питання
2/6	Тема 6. Аналіз вимог до ПЗ.	Вміти робити опис детальних вимог за допомогою діаграм діяльності та станів UML: діаграма діяльності, діаграма станів.	Питання Практична робота
2/7	Тема 7. Проектування програмного забезпечення.	Вміти будувати діаграму класів та об'єктів UML. Вміти проектувати програмне забезпечення. Вміти використовувати нотації UML для проектування ПЗ.	Питання Практична робота
2/6	Тема 8. Проектування ПЗ з допомогою структурних та поведінкових діаграм UML.	Вміти будувати діаграми: послідовності, пакетів, компонентів, розгортання.	Питання Практична робота
2/-	Тема 9. Якість ПЗ.	Знати основи якості програмного забезпечення, види якості. Знати моделі: якості програмних систем, Мак-Кола, ISO/IEC 9126. Знати стандартні показники якості та метрики якості.	Питання
2/4	Тема 10. Проектування UI.	Знати місце прототипу в життєвому циклі ПЗ, стандарти проектування людино-машинного інтерфейсу, засоби створення прототипів.	Питання Практична робота
2/-	Тема 11. Об'єктні моделі та реляційні бази даних.	Знати архітектурні рішення, функціональні проблеми. Вміти зчитувати дані та проводити взаємне відображення об'єктів та реляційних структур.	Питання
2/-	Тема 12. Чистий код.	Знати поняття чистого коду, стандарти створення чистого коду: змістовні імена, стандарти написання функцій, коментарів, класів. Вміти обробляти помилки.	Питання
2/4	Тема 13. Модульне тестування.	Знати призначення, переваги та характеристики якісного теста. Знати тестові двійники: об'єкт-заглушка, тестовая заглушка, тестовий агент, імітація, підставний об'єкт. Знати фреймворки тестування, підтримку модульного тестування на рівні мови.	Питання Практична робота
2/4	Тема 14. Системи контролю версій	Знати основи та особливості GIT: принципи збереження даних, стани файлу в робочому каталозі, процеси роботи з гілками, робочий процес одного розробника, робочий процес взаємодії декількох розробників, хостинг репозиторіїв, GitHub.	Питання Практична робота

2/-	Тема 15. Безперервна інтеграція та розгортання	Знати концепцію безперервної інтеграції Вміти використовувати Гаки (hooks) в системах контролю версій, інструменти складання, засоби безперервної інтеграції. Знати концепцію безперервного розгортання. Вміти використовувати засоби безперервного розгортання.	Питання
2/-	Тема 16. Принципи проектування	Знати шаблони DI: впровадження в конструктор, впровадження у властивість, впровадження в метод, навколишній контекст. Знати принципи SOLID: принцип єдиної відповідальності, принцип відкритості / закритості, принцип підстановки Лісков, принцип поділу інтерфейсів, принцип інверсії залежностей, принцип KISS, принцип YAGNI, принцип DRY, принцип TellDon't-Ask.	Питання
2/-	Тема 17. Архітектурні стилі	Знати архітектурні стилі їх поєднання. Знати архітектуру клієнт/сервера. Вміти проектувати на основі предметної області. Знати типи архітектурних стилів: багатошарова архітектура гексагональна архітектура, N-рівнева / 3-рівнева архітектура, об'єктно-орієнтована архітектура, сервісно-орієнтована архітектура та мікро-сервісна архітектура.	Питання
2/-	Тема 18. Багатошарова архітектура	Знати архітектуру базових типів застосунків, мобільні застосунків, Веб-застосунків. Знати особливості типових шарів рівень їх представлення. Вміти застосовувати шаблони реалізації бізнес логіки, шаблони реалізації шару доступу до даних.	Питання
2/-	Тема 19. Породжуючі шаблони проектування	Знати породжуючі шаблони заводський метод (Factory Method), абстрактна фабрика (Abstract Factory), одинак (Singleton), прототип (Prototype) та будівельник (Builder).	Питання
2/-	Тема 20. Структурні шаблони проектування	Знати структурні шаблони проектування: Декоратор (Decorator), Адаптер (Adapter), Фасад (Facade), Компоновщик (Composite). Заступник (Proxy), Міст (Bridge) та Пристосованець (Flyweight).	Питання
2/-	Тема 21. Поведінкові шаблони проектування	Знати поведінкові шаблони проектування: Стратегія (Strategy), Спостерігач (Observer). Команда (Command), Шаблонний метод (Template Method), Ітератор (Iterator), Стан (State), Ланцюжок Обов'язків (Chain of responsibility), Інтерпретатор (Interpreter), Посередник (Mediator), Хранитель (Memento) та Відвідувач (Visitor).	Питання
2/-	Тема 22. Супровід програмного забезпечення	Знати сутність та методи систем, завдання, та управління супроводом програмних систем.	Питання

Літературні джерела

1. Volker Gruhn Rüdiger Striemer The Essence of Software Engineering / Volker Gruhn Rüdiger Striemer. – Saint Philip Street Press, 2020. – 248 p.
2. Chhavi Raj Dosaj The Self-Taught Software Tester A Step By Step Guide to Learn Software Testing Using Real-Life Project / Chhavi Raj Dosaj. – Amazon Digital Services LLC - KDP Print US, 2020. – 217 p.
3. Liping Liu Requirements Modeling and Coding: An Object-Oriented Approach / L. Liping. – World Scientific Publishing Europe Ltd, 2020. – 451 p.
4. Постіл С. Д. UML. уніфікована мова моделювання інформаційних систем / С. Д. Постіл : Ун-т держ. фіск. служби України. - Ірпінь : Ун-т держ. фіск. служби України, 2019. - 321 с.
5. Мартін Р. Чистий код / Р. Мартін – М.: Фабула, 2019. – 416 с.
6. Роберт М. Чиста архітектура: мистецтво розробки програмного забезпечення» / Роберт Мартін, Фабула, 2019. – 416 с.
7. UML Tutorial [Електронний ресурс]. – Режим доступу: URL: <https://www.tutorialspoint.com/uml/index.htm>.
8. Java Tutorial [Електронний ресурс]. – Режим доступу: URL: <https://www.tutorialspoint.com/java/index.htm>.
9. C++ Tutorial [Електронний ресурс]. – Режим доступу: URL: <https://www.tutorialspoint.com/cplusplus/index.htm>.

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: Усі письмові роботи перевіряються на наявність плагіату і допускаються до захисту із коректними текстовими запозиченнями не більше 20%. Списування – заборонені (в т.ч. із використанням мобільних девайсів). Мобільні пристрої дозволяється використовувати лише під час он-лайн тестування.

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

Модуль 1		Модуль 2		Модуль 3	Модуль 4	Модуль 5
10 %	10 %	10 %	10 %	5 %	15%	40 %
Поточне оцінювання	Модульний контроль 1	Поточне оцінювання	Модульний контроль 2	Тренінг	Самостійна робота	Екзамен
Оцінка визначається як середнє арифметичне з оцінок, отриманих за виконання і захист 4 лабораторних	Модульна контрольна робота (15 тестових завдань та 1 практичне)	Оцінка визначається як середнє арифметичне з оцінок, отриманих за виконання і захист 4 лабораторних	Модульна контрольна робота (15 тестових завдань та 1 практичне)	Виконання завдань під час тренінгу (4 завдань)	Виконання завдання для самостійної роботи	25 тестових завдань та 2 практичних завдання

Шкала оцінювання:

За шкалою 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)