

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ЗАТВЕРДЖУЮ

Декан факультету Інформаційних технологій

Тетяна Говорущенко

Підпис

Декан факультету

2025 р.

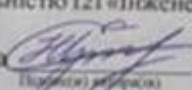
РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ Конструювання програмного забезпечення

Галузь знань
Спеціальність
Рівень вищої освіти
Освітньо-професійна програма
Обсяг дисципліни – 6 кредитів ЄКТС
Мова навчання
Статус дисципліни:

12 Інформаційні технології
121 «Інженерія програмного забезпечення»
Перший (бакалаврський)
Інженерія програмного забезпечення
Шифр дисципліни – ОФП.09
українська
обов'язкова (фахової підготовки)

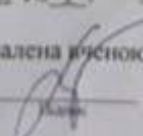
Форма здобуття освіти	Курс	Семестр	Обсяг дисципліни		Кількість годин						Форма семестрового контролю	
			Кредити ЄКТС	Години	Аудиторні заняття					Самостійна робота (в т.ч. ІРС)	Іспит	
					Разом	Лекції	Лабораторні роботи	Практичні заняття	Семінарські заняття			
Д	4	7	6	180	66	32	34			114		+

Робоча програма складена на основі освітньо-професійної програми «Інженерія програмного забезпечення» за спеціальністю 121 «Інженерія програмного забезпечення»

Робоча програма складена  к.пед.н., доцент, Наталія ПРАВОРСЬКА
Підпис (підпис) (підпис) Науковий ступінь, учено звання, ім'я, ПРІЗВИЩЕ, авторство

Схвалена на засіданні кафедри Інженерія програмного забезпечення

Протокол від 28 серпня 2025 р. № 1. Зав. кафедри  Леоніда БЕДРАТЮК
Підпис (підпис) Ім'я, ПРІЗВИЩЕ

Робоча програма розглянута та схвалена вченою радою факультету інформаційних
Голова вченої ради факультету  Тетяна ГОВОРУЩЕНКО
Підпис (підпис) Ім'я, ПРІЗВИЩЕ

Хмельницький 2025

2 ЛИСТ ПОГОДЖЕННЯ

Посада	Назва факультету	Підпис	Ініціали, прізвище
Завідувач кафедри інженерії програмного забезпечення, доктор, фіз.-мат. наук, проф.	Факультет інформаційних технологій		Леонід БЕДРАТЮК
Гарант освітньо-професійної програми, доктор, фіз.-мат. наук, проф.	Факультет інформаційних технологій		Леонід БЕДРАТЮК

КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Тип дисципліни
Рівень вищої освіти
Мова викладання

Семестр
Обсяг кредитів ЄКТС
Форма здобуття освіти

Обов'язкова
Перший
Українська
Сьомий
6,0
Денна/очна

Результати навчання. Відповідно до Стандарту вищої освіти та освітньої програми дисципліна має забезпечити:

- **компетентності:** здатність застосовувати знання у практичних ситуаціях; здатність працювати в команді; здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування; накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя; реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення; здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення.

- **програмні результати навчання:** аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки; уміти вибирати та використовувати відповідну задачу методологію створення програмного забезпечення; проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування; вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання; застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення; знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань; застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення; мати навички командної розробки, погодження, оформлення і випуску всіх видів програмної документації; вміти застосовувати методи компонентної розробки програмного забезпечення; знати та вміти застосовувати методи верифікації та валідації програмного забезпечення.

Зміст навчальної дисципліни. Елементи конструювання ПЗ. Ключові принципи конструювання. Інструменти конструювання. Мови опису, представлення, архітектурні каркаси. Техніка моделювання архітектури. Ідентифікація цілей та ключових сценаріїв. Системний підхід до розробки ПЗ. Моделі життєвого циклу ПЗ. Методологія створення ПЗ. Ідентифікація ключових проблем. Конструювання ПЗ та багатозадачність. Захисне конструювання ПЗ, синхронізація процесів, підходи при конструюванні ПЗ. Відладки при конструюванні ПЗ та контроль за ними.

Запланована навчальна діяльність: лекцій 32 год., лабораторних занять 34 год., самостійної роботи 114 год.; разом 180 год.

Методи навчання методи проблемного викладання, словесні, наочні (лекції); пояснювально-ілюстративні, проблемного викладання, дослідницькі, частково-пошукові (лабораторні заняття), проблемного викладання, дослідницькі, частково-пошукові (самостійна робота: індивідуальні завдання).

Форми і методи оцінювання результатів навчання: усне опитування, захист лабораторних робіт. письмові самостійні та контрольні роботи, письмовий іспит

Форма семестрового контролю: іспит – 7 семестр.

Навчальні ресурси:

1. Piotr Sliż. Organizacja procesowo-projektowa. Istota, modelowanie, pomiar dojrzałości. Difin – 2021 – 292 str.
2. Ian Sommerville. Software Engineering, 10th edition. Published by Pearson . July 14th 2021 – 816 p.
3. Кунгурцев О.Б., Новікова Н.О. Конструювання програмного забезпечення. Об'єктно-орієнтований підхід. Видавництво Кондор. – 2024 – 228 стор.
4. Andrew Hunt, David Thomas. The Pragmatic Programmer: Your Journey To Mastery, 20th Anniversary Edition (2nd Edition). Print2print – 2020 – 352 стор.
5. Tejas Chopra, Dharendra Sinha. System Design Guide for Software Professionals: Build scalable solutions. Packt Publishing, The Limited. – 2024 – pp.384
6. Модульне середовище для навчання MOODLE. Доступ до ресурсу: <https://msn.khnu.km.ua>.

Викладач: кандидат педагогічних наук, доцент Праворська Н.І.

3. Пояснювальна записка

Дисципліна «Конструювання програмного забезпечення» є обов'язковою дисципліною фахової підготовки і займає провідне місце у підготовці здобувачів першого (бакалаврського) рівня вищої освіти за освітньо-професійною програмою «Інженерія програмного забезпечення» в межах спеціальності 121 «Інженерія програмного забезпечення». У рамках дисципліни «Конструювання програмного забезпечення» вивчається процес конструювання програмних систем, розповсюдженні методики і практики побудови надійного програмного забезпечення.

Пререквізити – ОФП.01 Архітектура та проєктування програмного забезпечення, ОФП.02 Архітектура та проєктування програмного забезпечення (курсний проєкт), ОФП.10 Моделювання та оцінка програмного забезпечення.

Кореквізити – ОФП.17 Переддипломна практика, ОФП.18 Кваліфікаційна робота

Відповідно до освітньої програми дисципліна має забезпечити:

- **компетентності:** (ІК) Здатність розв'язувати складні спеціалізовані завдання або практичні проблеми інженерії програмного забезпечення, що характеризуються комплексністю та невизначеністю умов, із застосуванням теорій та методів інформаційних технологій; (ЗК02) Здатність застосовувати знання у практичних ситуаціях; (ЗК07) Здатність працювати в команді; (ФК02) Здатність брати участь у проєктуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування; (ФК10) Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя; (ФК11) Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення; (ФК12) Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення; (УК01) Здатність до паралельної роботи над етапами життєвого циклу програмного забезпечення та ефективної взаємодії з виконавцями різних етапів;

- **програмні результати навчання:** (ПРН01) Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки; (ПРН06) Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення; (ПРН08) Вміти розробляти людино-машинний інтерфейс; (ПРН10) Проводити передпроєктне обстеження предметної області, системний аналіз об'єкта проєктування; (ПРН11) Вибирати вихідні дані для проєктування, керуючись формальними методами опису вимог та моделювання; (ПРН12) Застосовувати на практиці ефективні підходи щодо проєктування програмного забезпечення; (ПРН13) Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань; (ПРН14) Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проєктування, тестування, візуалізації, вимірювань та документування програмного забезпечення; (ПРН16) Мати навички командної розробки, погодження, оформлення і випуску всіх видів програмної документації; (ПРН17) Вміти застосовувати методи компонентної розробки програмного забезпечення; (ПРН19) Знати та вміти застосовувати методи верифікації та валідації програмного забезпечення; (ПРН25) Вміти ефективно співпрацювати з виконавцями різних етапів життєвого циклу програмного забезпечення, координуючи одночасну роботу над етапами; (ПРН26) Знати та вміти застосовувати методики, інструменти та стратегії управління ресурсами, часом та комунікацією для забезпечення ефективної паралельної роботи над різними етапами життєвого циклу програмного забезпечення.

Мета дисципліни: формування теоретичних та практичних навичок в здобувачів першого (бакалаврського) рівня вищої освіти з питань проєктування та розробки інформаційних систем.

Предмет дисципліни – моделі життєвого циклу програмного забезпечення та основні методології та засоби розробки ПЗ. Технологія створення та документування програмних продуктів.

Завдання дисципліни: ознайомлення здобувачів першого (бакалаврського) рівня вищої освіти з сучасними методами та технологіями конструктивного програмного забезпечення; вивчення способів конструювання програмного забезпечення з використанням мови моделювання UML; формування вмінь та навиків виробітки конструкторських рішень; формування навичок роботи у сучасних інструментальних середовищах підтримки процесу конструювання програмних систем.

Результати навчання: аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки; уміти вибирати та використовувати відповідну задачі методологію створення програмного забезпечення; проводити передпроєктне обстеження предметної області,

системний аналіз об'єкта проектування; вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання; застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення; знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань; застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення; мати навички командної розробки, погодження, оформлення і випуску всіх видів програмної документації; вміти застосовувати методи компонентної розробки програмного забезпечення; знати та вміти застосовувати методи верифікації та валідації програмного забезпечення.

4. СТРУКТУРА ЗАЛІКОВИХ КРЕДИТІВ ДИСЦИПЛІНИ “Конструювання програмного забезпечення”

Теми	Лекції	Лаб. роб.	Самостійна робота здобувачів
Тема 1. Предмет і зміст дисципліни. Місце дисципліни в структурі навчання. Проблеми розробки ПЗ та шляхи їх вирішення. Інженерія ПЗ	2	2	14
Тема 2. Системний підхід до розробки ПЗ. Моделі життєвого циклу ПЗ. Методологія створення ПЗ.	4	4	14
Тема 3. Проектування при конструюванні ПЗ. Конструювання ПЗ та багатозадачність.	8	4	16
Тема 4. Захисне конструювання ПЗ, синхронізація процесів, підходи при конструюванні ПЗ (структурний та ООП)	4	4	16
Тема 5. Евристичні принципи конструювання програм та із захистом від помилок	2	4	14
Тема 6. Методи обробки помилок, аварійний захист ПЗ. Технологія налагодження ПЗ. Методи верифікації та валідації програмного забезпечення.	4	4	16
Тема 7. Відладки при конструюванні ПЗ та контроль за ними. Інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.	6	4	16
Тема 8. Моделювання роботи СТС Успадковане ПЗ. Мобільність ПЗ і реінженіринг ПЗ.	2	4	8
Разом за 7-й семестр	32	34	114

5. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ “Конструювання програмного забезпечення”

5.1. Зміст лекційного курсу*

№ лекції	Перелік тем лекцій, їх анотація	Кількість годин
Тема 1. Предмет і зміст дисципліни. Місце дисципліни в структурі навчання. Проблеми розробки ПЗ та шляхи їх вирішення. Інженерія ПЗ		
1	Лекція 1. Проблеми розробки ПЗ та шляхи їх вирішення. Інженерія ПЗ. Розв’язання професійних задач за допомогою сучасних досягнень науки і техніки. Програмування заданих прикладів і рішення задач за допомогою ПЗ – в чому різниця? Роль ПЗ і комп’ютерів у виробництві, соціальному житті і науці. Інженерія ПЗ. Проблеми розробки ПЗ та шляхи їх вирішення. Аналіз предметної області та пошук розв’язання задач проектування за допомогою сучасних досягнень науки і техніки. Передпроектне обстеження предметної області, системний аналіз об’єкта проектування, вибір необхідних для вирішення професійних завдань інформаційно-довідникові ресурсів з урахуванням сучасних досягнень науки і техніки. Література: [1] стор. 15-25, [2] стор 10-31, [3] стор. 5-18; [7]; [12] стор 7-12.	2
Тема 2. Системний підхід до розробки ПЗ. Моделі життєвого циклу ПЗ. Методологія створення ПЗ.		
2	Лекція 2. Якість ПЗ та фактори, які на нього впливають. Методи компонентної розробки програмного забезпечення. Технологія розробки ПЗ і якість ПЗ. Вибір методології створення програмного забезпечення.	2

	Характеристики якості ПЗ, які є важливими для користувача. Фактори, що впливають на якість ПЗ. Література: [2] стор 23-41; [5] стор 19-33; [13].	
3	Лекція 3. Системний підхід до розробки ПЗ. Каскадна модель життєвого циклу ПЗ. Конструювання ПЗ. Критерії якості ПЗ розробника. Стандарти та розробка ПЗ. Тимчасовий і «просторовий» аспекти системного підходу Етапи життєвого циклу ПЗ. Каскадна модель життєвого циклу. Конструювання ПЗ і внутрішні критерії якості ПЗ – критерії розробника. Стандарти по розробці ПЗ. Види і значення стандартів, вимоги стандартів. Вибір вихідних даних для проектування на основі формальних методів опису вимог та моделювання. Три групи процесів створення ПЗ. Література: [3] стор 49-58; [4] стор 53-69; [11], [14] стор 64-78.	2
Тема 3. Проектування при конструюванні ПЗ. Конструювання ПЗ та багатозадачність.		
4	Лекція 4. Верифікація ПЗ. V-образна схема життєвого циклу ПЗ. спіральна модель ЖЦ ПЗ. Технології розробки ПЗ: «важкі і легкі», методологія Agile, XP-програмування, DevOps (синхронне виконання етапів ЖЦ). Життєвий цикл ПЗ і процеси верифікації. Тестування, верифікація, валідація V-образна модель життєвого циклу ПЗ Спіральна модель ЖЦ ПО. «Важкі і швидкі» технології розробки ПЗ. Методологія Agile (робота в команді). Екстремальне (XP) програмування. Технології DevOps (інструменти та стратегії управління ресурсами, часом та комунікацією – синхронне виконання етапів ЖЦ). Література: [4] стор 69-74; [7]; [8]; [10], [15] стор. 73-88.	2
5	Лекція 5. Три види програмних розробок з точки зору конструювання, технології створення і експлуатації. Принципи класифікації ПЗ. Три види програмних розробок з точки зору конструювання, технології створення і експлуатації. Види документів, що випускаються на ПЗ, по етапах розробки системи. Ефективні підходи щодо проектування програмного забезпечення. Методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань. Командна розробка, погодження, оформлення і випуск всіх видів програмної документації. Література: [2] стор 91-112; [4] стор. 132-148; [5] стор. 98-135; [6] стор. 131-157; [16] стор. 54-87.	2
6	Лекція 6. Проектування архітектури ПЗ. Ітеративний характер проектування системи і ПЗ. Ціна помилок проектування. CASE технології розробки ПЗ. Ітеративний характер проектування системи і ПЗ. Стадії проектування. Завдання, які вирішуються на різних стадіях проектування системи і ПЗ. Ціна помилок проектування. Проектування, засноване на моделюванні. CASE технології розробки ПЗ. Завдання і результати архітектурного проектування ПЗ. Технологія RationalRose, UML. Література: [1] стор. 154-187; [2] стор. 148-173; [3] стор. 1134-156; [7]; [16] стор. 130-149. Структурна схема ПЗ. СТС, як засіб конструювання ПЗ. Ієрархічна структура ПЗ СТС. Циклічність (періодичність) у часі вирішення завдань ПЗ. Співпраця між виконавцями різних етапів ЖЦ ПЗ. Структура системи, ієрархія управління і структура ПЗ. Розвиток системи і проведення змін в ієрархічній структурі ПЗ. Циклічність (періодичність) у часі вирішення завдань управління і роботи ПЗ. Співпраця команд розробки та команд операцій на різних етапах ЖЦ при розробці ПЗ. Література: [4] стор. 321-388; [5] стор. 269-301; [6] стор.308-349.	2
7	Лекція 7. Тимчасова діаграма роботи системи, як засіб конструювання ПЗ, яке забезпечує паралельні фізичні процеси. Багатозадачна робота ПЗ і її причини. Тимчасова діаграма роботи системи і ПЗ з паралельними фізичними процесами. Проблема поділу ресурсів для безпечної багатозадачною роботи ПО СТС. Причини багатозадачності. Багатозадачна робота ПЗ і її реалізації при колективній розробці. Завдання, процеси і потоки. Контекст процесу. Узагальнена схема варіантів спільного	2

	використання інформації взаємодіючими процесами. Реалізація багатозадачності за рахунок паралельних обчислень. Технологія Open MP. Закон Амдела Література: [1] стор. 187-195; [4] стор. 281-310; [13].	
Тема 4. Захисне конструювання ПЗ, синхронізація процесів, підходи при конструюванні ПЗ (структурний та ООП)		
8	Лекція 8. Захисне конструювання ПЗ з паралельними процесами. Задачі синхронізації процесів. Критичний ресурс ЦОМ. Основне правило захисту критичних ресурсів ЦОМ при конструюванні ПЗ. Завдання синхронізації процесів. Тимчасова і логічна синхронізація. Взаємне виключення процесів. Використання м'ютексів. Завдання синхронізації «Читачі-письменники» Завдання синхронізації. «Обідають філософи» Система IntelThreadChecker (ITC) і типи виявляються нею помилок. Література: [3] стор. 194-212; [4] стор. 315-342; [6] стор. 267-298; [14] стор. 312-354.	2
9	Лекція 9. Конструювання ПЗ з мінімізацією його складності. Основні поняття структурного та об'єктно-орієнтованого підходу до конструювання ПЗ. Проектування «зверху вниз» і «знизу вгору». Характеристики ПЗ, які поліпшують процеси його розробки і супроводу. Мінімізація складності ПЗ. Стандартні прийоми в конструюванні. Основні поняття структурного і об'єктно-орієнтованого підходу до конструювання ПЗ. Рефакторинг. Особливості конструювання програм для вбудованих ЦОМ критичних систем. Фіксований розподіл пам'яті. Проектування знизу-вгору і проектування зверху-вниз. Програмні заглушки і їх використання Література: [3] 213-247; [5] стор. 312-368; [7].	2
Тема 5. Евристичні принципи конструювання програм та із захистом від помилок		
10	Лекція 10. Евристичні принципи конструювання програм. Конструювання ПЗ, пристосованого до змін (готового до проведення змін). Класи реального світу предметної області і штучні об'єкти. Надмірно великі і неправильно названі класи. Евристичні прийоми конструювання методів, запобігання дублювання коду. Приховування інформації. Дві категорії секретів програм. Надмірне поширення інформації в програмі Захист від переривання програми в «незручний» час Підвищення пристосованості ПЗ до змін (готовності до змін). Література: [1] стор. 35-67, стор. 312-355; [2] стор. 249-297; [7]; [12] стор. 25-37.	2
Тема 6. Методи обробки помилок, аварійний захист ПЗ. Технологія налагодження ПЗ. Методи верифікації та валідації програмного забезпечення.		
11	Лекція 11. Конструювання програм із захистом від помилок (стійких до помилок програм). Низькорівневий і високорівневий захист від помилок в програмах. Стійкі до помилок програми Види контролю роботи ПЗ. Контроль роботи ПЗ вбудованими засобами без припинення його функціонування. Еталони для контролю роботи ПЗ. Низькорівневі засоби виявлення помилкового функціонування ПЗ. Виняткові ситуації (Винятки). Література: [5] стор. 369-379; [6] стор. 319-364, [14] стор. 351-369; [15] стор. 158-176.	2
12	Лекція 12. Методи обробки помилок у вхідних і вихідних даних. Твердження і загальні принципи їх використання Конструювання аварійного захисту в ПЗ для мінімізації збитку від проявилися помилок. Три основні кроки захисного конструювання. Методи обробки можливих помилок у вхідних і вихідних даних. Що важливіше стійкість або коректність? Твердження і загальні принципи їх використання. Системний рівень захисту від помилок при конструюванні ПЗ. Стратегії безпеки. Три типи реакції ПЗ на виявлену помилку. Відмовостійкі системи. Перелік позаштатних ситуацій. Аварійний захист. Захисне конструювання ПЗ зі скасуванням помилкових команд, виданих з ПЗ або на ПЗ. Література: [3] стор. 67-111; [4] стор. 412-469, [18].	2

Тема 7. Відладки при конструюванні ПЗ та контроль за ними. Інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.		
13	Лекція 13. Технологія налагодження ПЗ. Помилки ПЗ. Статична, динамічна, структурна, функціональна відладки. Помилки ПЗ, відладка і тестування програмного забезпечення. Два інструментальних засоби відладки. Аналіз виявлених в ПЗ помилок і їх класифікація Статична відладка і динамічна відладка Принцип «білого» і «чорного» ящика при динамічній відладці ПЗ. Функціональна відладка. Література: [7], [9] [17] стор. 412-531, [18].	2
14	Лекція 14. Структурна динамічна відладка. Автономна відладка (АВ) і комплексна відладка (КВ) ПЗ. Послідовність дій при відладці ПЗ. Структурна динамічна відладка. Вибір критеріїв для відбору варіантів роботи ПЗ при відладці Автономна відладка (АВ) і комплексна відладка (КВ) ПЗ. Драйвери і заглушки при автономній відладці. Послідовність дій при відладці ПЗ. Література: [4] стор. 536-598, [10], [21].	2
15	Лекція 15. Деякі проектні моделі оцінки числа маршрутів при відладці ПЗ. Контроль відладки ПЗ в процесі налагодження. Наближений метод оцінки числа варіантів для відладки ПЗ. Графи-дерева, як модель структури ПЗ. Регулярне і випадкове дерево структури ПЗ і стійкість його структурного параметра. Контроль налагодженості ПЗ в процесі відладки. Гіпотеза Джелінського-Моранді і математична модель надійності ПЗ. Приріст кількості помилок, виявлених на інтервалі. Метод найменших квадратів для апроксимації експериментальних даних по помилках ПЗ. Література: [4] стор. 611-632, [11], [17] стор. 346-451.	2
Тема 8. Моделювання роботи СТС Успадковане ПЗ. Мобільність ПЗ і реінженіринг ПЗ.		
16	Лекція 16. Моделювання роботи СТС з метою генерації даних для проведення комплексної відладки ПЗ. Успадковане ПЗ. Мобільність ПЗ і реінжиніринг ПЗ. Проблема генерації даних на комплексну відладку. Принципи математичного моделювання. Переваги математичного моделювання зовнішнього середовища. Успадковане ПЗ. Мобільність ПЗ і реінжиніринг ПЗ. Емуляція роботи старої ЦОМ на новій апаратній платформі. Література: [5] стор. 342-387, [7], [11]; [22].	2
Разом за 7-й семестр		32

5.2 Зміст лабораторних занять

№ з/п	Теми лабораторних занять	Години
1	Планування розробки ПЗ. Етапи розробки програмного забезпечення при структурному підході до програмування. Стадія «Технічне завдання». Передпроектне обстеження предметної області, системний аналіз об'єкта проектування.	2
2	Структурний підхід до програмування. Стадія «Ескізний проект». Сітковий графік виконання робіт. Вибір методології розробки ПЗ. Опис вимог до ПЗ. Робота в команді.	4
3	Стадія розробки програмного забезпечення «Технічний проект». Структурний підхід до програмування. Розробка людино-машинного інтерфейсу. Розробка прототипу ПЗ.	4
4	Етапи розробки програмного забезпечення Стадія «Реалізація». Проектування та реалізація проекту ПЗ з інтерфейсом користувача.	4
5	Виконання оцінки проекту на основі loc- і fr-метрик.	4
6	Аналіз чутливості програмного проекту на основі моделі	6

	COCOMO II	
7	Метрики об'єктно-орієнтованих програмних систем.	6
8	Організація таблиць у трансляторах і робота з ними. Рефакторинг. Оптимізація коду.	4
Разом за 7-й семестр		34

5.3 Зміст самостійної роботи

Об'єм самостійної роботи з дисципліни **“Конструювання програмного забезпечення”** становить 114 години. Вони включають опрацювання лекційного матеріалу, теоретичних і лабораторних завдань, підготовку до виконання лабораторних робіт, їх захисту і поточне тестування.

Номер тижня	Назва теми	Години
1	Тема 1. Предмет і зміст дисципліни. Місце дисципліни в структурі навчання. Проблеми розробки ПЗ та шляхи їх вирішення. Інженерія ПЗ. Опрацювання лекційного матеріалу, підготовка до виконання лабораторної роботи №1. Література: [1] стор. 15-25, [2] стор 10-31, [3] стор. 5-18; [7]; [12] стор 7-12.	6
2	Тема 1. Предмет і зміст дисципліни. Місце дисципліни в структурі навчання. Проблеми розробки ПЗ та шляхи їх вирішення. Інженерія ПЗ. Опрацювання лекційного матеріалу. Виконання та захист лабораторної роботи № 1. Література: [1] стор. 15-25, [2] стор 10-31, [3] стор. 5-18; [7]; [12] стор 7-12.	8
3	Тема 2. Системний підхід до розробки ПЗ. Моделі життєвого циклу ПЗ. Методологія створення ПЗ. Опрацювання лекційного матеріалу, підготовка та виконання лабораторної роботи №2. Література: [2] стор 23-41; [5] стор 19-33; [13].	6
4	Тема 2. Системний підхід до розробки ПЗ. Моделі життєвого циклу ПЗ. Методологія створення ПЗ. Опрацювання лекційного матеріалу. Захист лабораторної роботи №2. Література: [3] стор 49-58; [4] стор 53-69; [11], [14] стор 64-78.	8
5	Тема 3. Проектування при конструюванні ПЗ. Конструювання ПЗ та багатозадачність. Опрацювання лекційного матеріалу. Підготовка та виконання лабораторної роботи №3. Література: [4] стор 69-74; [7]; [8]; [10], [15] стор. 73-88.	4
6	Тема 3. Проектування при конструюванні ПЗ. Конструювання ПЗ та багатозадачність. Опрацювання лекційного матеріалу, захист лабораторної роботи №3. Література: [2] стор 91-112; [4] стор. 132-148; [5] стор. 98-135; [6] стор. 131-157; [16] стор. 54-87.	6
7	Тема 3. Проектування при конструюванні ПЗ. Конструювання ПЗ та багатозадачність. Опрацювання лекційного матеріалу. Підготовка та виконання лабораторної роботи №4. Література: [1] стор. 154-187; [2] стор. 148-173; [3] стор. 1134-156; [7]; [16] стор. 130-149. Підготовка до тестування	6
8	Тема 4. Захисне конструювання ПЗ, синхронізація процесів, підходи при конструюванні ПЗ (структурний та ООП). Опрацювання лекційного матеріалу, захист лабораторної роботи №4. Література: [3] стор. 194-212; [4] стор. 315-342; [6] стор. 267-298; [14] стор. 312-354.	6
9	Тема 4. Захисне конструювання ПЗ, синхронізація процесів, підходи при конструюванні ПЗ (структурний та ООП). Опрацювання лекційного матеріалу, підготовка та виконання лабораторної роботи №5. Література: [3] 213-247; [5] стор. 312-368; [7].	8

10	Тема 5. Евристичні принципи конструювання програм та із захистом від помилок. Опрацювання лекційного матеріалу, захист лабораторної роботи №5. Література: [1] стор. 35-67, стор. 312-355; [2] стор. 249-297; [7]; [12] стор. 25-37.	6
11	Тема 5. Евристичні принципи конструювання програм та із захистом від помилок. Опрацювання лекційного матеріалу, підготовка та виконання лабораторної роботи №6. Література: [1] стор. 35-67, стор. 312-355; [2] стор. 249-297; [7]; [12] стор. 25-37.	8
12	Тема 6. Методи обробки помилок, аварійний захист ПЗ. Технологія налагодження ПЗ. Методи верифікації та валідації програмного забезпечення. Опрацювання лекційного матеріалу, захист лабораторної роботи №6. Література: [5] стор. 369-379; [6] стор. 319-364, [14] стор. 351-369; [15] стор. 158-176.	8
13	Тема 6. Методи обробки помилок, аварійний захист ПЗ. Технологія налагодження ПЗ. Методи верифікації та валідації програмного забезпечення. Опрацювання лекційного матеріалу, підготовка та виконання лабораторної роботи №7. Література: [3] стор. 67-111; [4] стор. 412-469, [18].	8
14	Тема 7. Відладки при конструюванні ПЗ та контроль за ними. Інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення. Опрацювання лекційного матеріалу. Захист лабораторної роботи №7. Література: [4] стор. 536-598, [7], [9]; [10], [17] стор. 412-531, [18]; [21].	8
15	Тема 7. Відладки при конструюванні ПЗ та контроль за ними. Інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення. Опрацювання лекційного матеріалу, підготовка та виконання лабораторної роботи №8. Література: [4] стор. 611-632, [11], [17] стор. 346-451.	8
16	Тема 8. Моделювання роботи СТС Успадковане ПЗ. Мобільність ПЗ і реінженіринг ПЗ. Опрацювання лекційного матеріалу, захист лабораторної роботи №8. Література: [5] стор. 342-387, [7], [11]; [22]. Підготовка до іспиту	8
Разом за 7-й семестр		114

На самостійне опрацювання студентів виносяться визначені у методичних рекомендаціях до лабораторних занять та самостійної роботи питання з кожної теми. Керівництво самостійною роботою та контроль за виконанням індивідуального завдання здійснюється викладачем згідно з розкладом консультацій у позаурочний час.

Вимоги до виконання контрольної роботи (для студентів заочної форми здобуття освіти) та індивідуального домашнього завдання (для студентів денної форми здобуття освіти) викладені в Модульному середовищі для навчання на сторінці навчальної дисципліни.

6. Технології та методи навчання

Процес навчання з дисципліни ґрунтується на використанні традиційних та сучасних технологій та методів навчання, зокрема: методи навчання за джерелом передачі і сприймання інформації (словесні (пояснення, дискусія, консультування), практичні (інструктування, розв'язування ситуаційних задач), наочні (демонстрування, ілюстрування, спостереження); за логікою передачі і сприймання навчальної інформації; за рівнем самостійності пізнавальної діяльності (методи проблемного викладу, частково пошукові, дослідницькі); методи стимулювання і мотивації учіння, інтерактивні; метод аналізу конкретних ситуацій (case-study) з використанням технологій візуалізації, інформаційно-комунікаційних та технології дистанційного навчання (сервіс для проведення онлайн конференцій Zoom, Модульне середовище для навчання тощо).

Процес навчання з дисципліни ґрунтується на використанні традиційних та сучасних технологій, зокрема:

- лекції (з використанням мультимедійних презентацій, методів візуалізації, пояснення, проблемного й інтерактивного навчання, методів стимулювання і мотивації, **інформаційно-комунікаційних технологій**, інтенсифікації та індивідуалізації навчання, тощо);
- лабораторні заняття (з використанням методів комп'ютерного моделювання, методів проєктної діяльності, тренінгових вправ, аналіз проблемних ситуацій, пояснення, дискусія, бесіда, демонстрування, спостереження, з використанням кейсів, розв'язування задач, презентацій);
- самостійна робота (робота над засвоєнням теоретичного матеріалу, виконання індивідуальних та домашніх завдань, підготовка до поточного та підсумкового контролю тощо).

7. Методи контролю

Поточний контроль здійснюється під час лабораторних занять, а також у дні проведення контрольних заходів, встановлених робочою програмою і графіком освітнього процесу.

При цьому використовуються такі методи поточного контролю:

- тестовий контроль теоретичного матеріалу;
- оцінювання результатів роботи на лабораторних заняттях (розв'язування задач, участь у обговоренні ситуацій, самостійні роботи).

При виведенні підсумкової семестрової оцінки враховуються результати як поточного контролю, так і підсумкового контролю, який проводиться з усього матеріалу дисципліни за білетами, попередньо розробленими і затвердженими на засіданні кафедри. Здобувач вищої освіти, який набрав з будь-якого виду навчальної роботи, суму балів нижчу за 60 відсотків від максимального балу, не допускається до семестрового контролю, поки не виконає обсяг роботи, передбачений Робочою програмою. Здобувач вищої освіти, який набрав позитивний середньозважений бал (60 відсотків і більше від максимального балу) з усіх видів поточного контролю і не склав іспит, вважається таким, який має академічну заборгованість. Ліквідація академічної заборгованості із семестрового контролю здійснюється у період екзаменаційної сесії або за графіком, встановленим деканатом відповідно до «Положення про контроль і оцінювання результатів навчання здобувачів вищої освіти у ХНУ».

8. Політика дисципліни

Політика навчальної дисципліни загалом визначається системою вимог до здобувача вищої освіти, що передбачені чинними положеннями Університету про організацію і навчально-методичне забезпечення освітнього процесу. Зокрема, проходження інструктажу з техніки безпеки; відвідування занять з дисципліни є обов'язковим. За об'єктивних причин (підтверджених документально) теоретичне навчання за погодженням із лектором може відбуватись в індивідуальному режимі. Успішне опанування дисципліни і формування фахових компетентностей і програмних результатів навчання передбачає необхідність підготовки до практичних занять (вивчення теоретичного матеріалу з теми, активно працювати на занятті, розв'язувати задачі, брати участь у дискусіях щодо прийнятих рішень при виконанні здобувачами задач).

Здобувачі вищої освіти зобов'язані дотримуватися термінів виконання усіх видів робіт у встановлені терміни, передбачених робочою програмою навчальної дисципліни. Пропущене практичне заняття здобувач зобов'язаний відпрацювати у встановлений викладачем термін, але не пізніше, ніж за два тижні до кінця теоретичних занять у семестрі.

Засвоєння студентом теоретичного матеріалу з дисципліни оцінюється за результатами опитування під час практичних занять, тестування й виконання індивідуального домашнього завдання. Виконання індивідуального завдання завершується його здачею на перевірку у терміни, встановлені графіком самостійної роботи. У якості ІДЗ здобувач може підготувати реферат або тези доповіді на конференцію за однією з тем навчальної дисципліни при дотриманні узгоджених з викладачем термінів його виконання.

Здобувач вищої освіти, виконуючи самостійну роботу з дисципліни, має дотримуватися політики доброчесності (заборонені списування, підказки, плагіат, використання штучного інтелекту (без вірного цитування)). У разі порушення політики академічної доброчесності в будь-яких видах навчальної роботи здобувач вищої освіти отримує незадовільну оцінку і має повторно виконати

завдання з відповідної теми (виду роботи), що передбачені робочою програмою. Будь-які форми порушення академічної доброчесності під час вивчення навчальної дисципліни не допускаються та не толеруються.

У межах вивчення навчальної дисципліни здобувачам вищої освіти передбачено визнання і зарахування результатів навчання, набутих шляхом неформальної освіти, що розміщені на доступних платформах (<https://github.com>, <https://docs.oracle.com/javase/tutorial/java/index.html>, <http://www.bruceeckel.com/>, <http://java.sun.com/products/jdk/1.3/docs/index.html>, <http://java.sun.com/products/jdk/1.3/devdocs-vs-specs.html>, <http://developer.java.sun.com/developer/onlineTraining/>, <http://java.sun.com/products/java-media/jmf/2.1.1/guide/JMFTOC.html>), які сприяють формуванню компетентностей і поглибленню результатів навчання, визначених робочою програмою дисципліни, або забезпечують вивчення відповідної теми та/або виду робіт з програми навчальної дисципліни (детальніше у Положенні про порядок визнання та зарахування результатів навчання здобувачів вищої освіти у ХНУ).

9. Оцінювання результатів навчання студентів у семестрі

Оцінювання академічних досягнень здобувача вищої освіти здійснюється відповідно до «Положення про контроль і оцінювання результатів навчання здобувачів вищої освіти у ХНУ». При поточному оцінюванні виконаної здобувачем роботи з кожної структурної одиниці і отриманих ним результатів викладач виставляє йому певну кількість балів із призначених робочою програмою для цього виду роботи. При цьому кожна структурна одиниця (робота) може бути зарахована, якщо здобувач набрав не менше 60 відсотків (мінімальний рівень для позитивної оцінки) від максимально можливої суми балів, призначеної структурній одиниці.

Будь-які форми порушення академічної доброчесності **не допускаються та не толеруються**.

Отриманий здобувачем бал за зарахований вид навчальної роботи (структурну одиницю) після її оцінювання викладач виставляє в електронному журналі обліку успішності здобувачів вищої освіти. За умови виконання усіх видів навчальної роботи за результатами поточного контролю протягом вивчення навчальної дисципліни, встановлених її Робочою програмою, здобувач денної форми здобуття освіти з навчальної дисципліни, підсумковим контролем для якої є іспит, може набрати до 60 балів (здобувач заочної форми – до 50 балів). Позитивну підсумкову оцінку здобувач може отримати, якщо за результатами поточного та підсумкового контролів набере від 60 до 100 балів. Семестрова підсумкова оцінка розраховується в автоматизованому режимі в інформаційній підсистемі «Електронний журнал» (ІС «Електронний університет») і відповідно до накопиченої суми балів визначається оцінка за інституційною шкалою та шкалою ЄКТС (див. таблицю Співвідношення...), яка заноситься в екзаменаційну відомість, а також до Індивідуального навчального плану здобувача вищої освіти.

Таблиця – Критерії оцінювання навчальних досягнень здобувача вищої освіти

Оцінка та рівень досягнення здобувачем запланованих ПРН та сформованих компетентностей	Узагальнений зміст критерія оцінювання
Відмінно (високий)	Здобувач вищої освіти глибоко і у повному обсязі опанував зміст навчального матеріалу, легко в ньому орієнтується і вміло використовує понятійний апарат; уміє пов'язувати теорію з практикою, вирішувати практичні завдання, впевнено висловлювати і обґрунтовувати свої судження. Відмінна оцінка передбачає логічний виклад відповіді мовою викладання (в усній або у письмовій формі), демонструє якісне оформлення роботи і володіння спеціальними приладами та інструментами, прикладними програмами. Здобувач не вагається при видозміні запитання, вміє робити детальні та узагальнюючі висновки, демонструє практичні навички з вирішення фахових завдань. При відповіді допустив дві–три несуттєві <i>помилки</i> .

Добре (середній)	Здобувач вищої освіти виявив повне засвоєння навчального матеріалу, володіє понятійним апаратом, орієнтується у вивченому матеріалі; свідомо використовує теоретичні знання для вирішення практичних задач; виклад відповіді грамотний, але у змісті і формі відповіді можуть мати місце окремі неточності, нечіткі формулювання правил, закономірностей тощо. Відповідь здобувача вищої освіти будується на основі самостійного мислення. Здобувач вищої освіти у відповіді допустив дві–три <i>несуттєві помилки</i> .
Задовільно (достатній)	Здобувач вищої освіти виявив знання основного програмного матеріалу в обсязі, необхідному для подальшого навчання та практичної діяльності за професією, справляється з виконанням практичних завдань, передбачених програмою. Як правило, відповідь здобувача вищої освіти будується на рівні репродуктивного мислення, здобувач вищої освіти має слабкі знання структури навчальної дисципліни, допускає неточності і <i>суттєві помилки</i> у відповіді, вагається при відповіді на видозмінене запитання. Разом з тим, набув навичок, необхідних для виконання нескладних практичних завдань, які відповідають мінімальним критеріям оцінювання і володіє знаннями, що дозволяють йому під керівництвом викладача усунути неточності у відповіді.
Незадовільно (недостатній)	Здобувач вищої освіти виявив розрізнені, безсистемні знання, не вміє виділяти головне і другорядне, допускається помилок у визначенні понять, перекручує їх зміст, хаотично і невпевнено викладає матеріал, не може використовувати знання при вирішенні практичних завдань. Як правило, оцінка «незадовільно» виставляється здобувачеві вищої освіти, який не може продовжити навчання без додаткової роботи з вивчення навчальної дисципліни.

Структурування дисципліни за видами навчальної роботи і оцінювання результатів навчання студентів денної форми здобуття освіти у семестрі

Аудиторна робота								Контрольні заходи		Семест- ровий контроль	Разом
<i>Другий семестр</i>											
Лабораторні роботи №:								Тестовий контроль:		Іспит	Сума балів
1	2	3	4	5	6	7	8	T1	T2		
Кількість балів за вид навчальної роботи (мінімум-максимум)											
3-5	3-5	3-5	3-5	3-5	3-5	3-5	3-5	3-5	3-5		
24-40								6-10	6-10	24-40	60-100

Умовні позначення: Т – тема навчальної дисципліни

Оцінювання результатів захисту лабораторної роботи

Виконана й оформлена відповідно до встановлених Методичними рекомендаціями вимог лабораторна робота комплексно оцінюється викладачем при її захисті з урахуванням таких критеріїв: самостійність та правильність виконання; повнота відповіді та знання методики проектування та розробки коду (застосунку); наявність відповідей на контрольні питання до відповідної лабораторної роботи, наявність звіту зі скріпками коду та результатів його роботи, тощо.

Результат виконання і захисту здобувачем вищої освіти кожної лабораторної роботи оцінюється відповідно до таблиці Критеріїв оцінювання навчальних досягнень здобувача вищої освіти та рівня досягнення здобувачем запланованих ПРН та сформованих компетентностей з присвоєнням йому відповідної суми балів.

У випадку виявлення здобувачем рівня знань, нижчого ніж 60 відсотків від максимального балу, встановленого Робочою програмою для кожної структурної одиниці, лабораторна робота йому не зараховується і для її захисту він має детальніше опрацювати матеріал з теми роботи, методику її виконання, виправити грубі помилки та повторно вийти на її захист у призначений для цього викладачем час.

Оцінювання результатів навчання з теоретичного матеріалу (тестовий контроль)

Тест1, передбачений Робочою програмою, складається із 30 тестових завдань, кожне з яких є рівнозначним. Максимальна сума балів, яку може набрати студент, складає 100 (*кількість набраних балів за тестове завдання може бути різною*).

Відповідно до таблиці структурування видів робіт за тематичний контроль здобувач залежно від кількості правильних відповідей може отримати від 60 до 100 балів.

Розподіл балів в залежності від наданих правильних відповідей на тестові завдання

Кількість правильних відповідей	1-17	18-23	24-26	27-30
Відсоток правильних відповідей	0-59	60-79	80-89	90-100
Отримана оцінка	-	3	4	5
Набрана кількість балів	>6	6	8	10

На тестування відводиться 30 хвилин. Студент проходить тестування в он-лайн режимі у Модульному середовищі для навчання. При отриманні негативної оцінки тест слід перездати до терміну *наступного* контролю.

Тест2, передбачений Робочою програмою, відбувається за допомогою системи Strum, розробленою компанією Genesis в рамках вивчення дисципліни «Створення та розвиток ІТ-продуктів»

Розподіл балів в залежності від наданих правильних відповідей на тестові завдання

Кількість пройдених модулів	0-2	3-4	5-9	10-12
Відсоток правильних відповідей	10%	25%	50%	100%
Отримана оцінка	-	3	4	5 та отримано сертифікат
Набрана кількість балів	>6	6	8	10

Оцінювання результатів підсумкового семестрового контролю (іспит)

Освітня програма передбачає підсумковий семестровий контроль з дисципліни у формі іспиту, завданням якого є системне й об'єктивне оцінювання як теоретичної, так і практичної підготовки здобувача з навчальної дисципліни. Складання іспиту відбувається за попередньо розробленими і затвердженими на засіданні кафедри білетами. Відповідно до цього в екзаменаційному білеті пропонується поєднання питань як теоретичного (в т.ч. у тестовій формі), так і практичного характеру.

Таблиця – Оцінювання результатів підсумкового семестрового контролю здобувачів денної форми навчання (40 балів для підсумкового контролю)

Види завдань	Для кожного окремого виду завдань		
	Мінімальний (достатній) бал (задовільно)	Потенційні позитивні бали* (середній бал) (добре)	Максимальний (високий) бал (відмінно)
Теоретичне питання № 1	6	4	5
Теоретичне питання № 2	6	4	5
Практичне завдання	6	24	30
Разом:	24		40

Примітка. *Позитивний бал за іспит, відмінний від мінімального (24 бали) та максимального (40 балів), знаходиться в межах 25-39 балів та розраховується як сума балів за усі структурні елементи (завдання) іспиту.

Для кожного окремого виду завдань підсумкового семестрового контролю застосовуються критерії оцінювання навчальних досягнень здобувача вищої освіти, наведені вище (**Таблиця – Критерії оцінювання навчальних досягнень здобувача вищої освіти**).

Семестровий іспит виставляється, якщо загальна сума балів, яку набрав студент з дисципліни за результатами поточного контролю, знаходиться у межах від 60 до 100 балів. При цьому за

інституційною шкалою ставиться оцінка «відмінно/добре/задовільно», а за шкалою ЄКТС – буквене позначення оцінки, що відповідає набраній студентом сумі балів відповідно до таблиці Співвідношення.

Таблиця – Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС

Оцінка ЄКТС	Рейтингова шкала балів	Інституційна оцінка (рівень досягнення здобувачем вищої освіти запланованих результатів навчання з навчальної дисципліни)	
		Залік	Іспит/диференційований залік
A	90-100	Зараховано	Відмінно/Excellent – високий рівень досягнення запланованих результатів навчання з навчальної дисципліни, що свідчить про безумовну готовність здобувача до подальшого навчання та/або професійної діяльності за фахом
B	83-89		Добре/Good – середній (максимально достатній) рівень досягнення запланованих результатів навчання з навчальної дисципліни та готовності до подальшого навчання та/або професійної діяльності за фахом
C	73-82		Задовільно/Satisfactory – Наявні мінімально достатні для подальшого навчання та/або професійної діяльності за фахом результати навчання з навчальної дисципліни
D	66-72		
E	60-65		
FX	40-59	Незараховано	Незадовільно/Fail – Низка запланованих результатів навчання з навчальної дисципліни відсутня. Рівень набутих результатів навчання є недостатнім для подальшого навчання та/або професійної діяльності за фахом
F	0-39		Незадовільно/Fail – Результати навчання відсутні

10. Питання до тестового контролю студентів з дисципліни «Конструювання програмного забезпечення»

1. Формула конструювання програмного забезпечення
2. Для запису будь-якої програми в принципі досить тільки трьох конструкцій керування:
3. Формальні методи допомагають
4. Що відноситься до основних етапів розробки ПЗ:
5. Процес конструювання ПЗ – це
6. Засоби для створення додатків – це
7. Системи програмування включають
8. Case-Технологія – це
9. Життєвий цикл ПЗ – це
10. При реалізації проекту необхідно здійснювати наступні види діяльності відповідно до прийнятих в організації політиками і процедурами відносно процесу реалізації програмних засобів:
11. У результаті успішного здійснення процесу реалізації програмних засобів:
12. Виконавець повинен оцінювати програмний код і результати випробувань, враховуючи наступні критерії:
13. Моделі життєвого циклу ПЗ можуть використовуватися для представлення
14. Переваги застосування каскадної моделі полягають у наступному:
15. Основний принцип V- Образної моделі полягає в тому
16. При використанні спіральної моделі прикладне ПЗ створюється
17. В спіральній моделі Боем формулює 10 найпоширеніших (по пріоритетах) ризиків
18. Прогнозовані (предикативні) методології фокусуються на
19. RUP –
20. MSF складається з
21. MSF включає в себе ряд основних принципів:
22. При використанні XP
23. Назвати методи/практики, притаманні Crystal Clear

24. Основні характеристики Crystal Clear:
25. Якість –
26. Керування якістю – це
27. Сукупність вимірюваних характеристик, одиниць виміру, вимірювальних шкал і зв'язків, установлених між ними – це
28. Методи засновані на отриманні інформації про характеристики програмного забезпечення під час випробувань або функціонування шляхом реєстрації й підрахунку певних подій – це
29. Методи засновані на отриманні інформації про характеристики програмного забезпечення шляхом їхнього сприйняття органами почуттів – це
30. Ієрархію характеристик якості утворюють

11. Питання для самоперевірки та іспиту з дисципліни “Конструювання програмного забезпечення”

1. Каскадні моделі життєвого циклу розробки програмних систем
2. Діаграми пакетів в UML
3. Спіралевидні моделі життєвого циклу розробки програмних систем
4. Способи задання обмежень в UML діаграмах.
5. Спіралевидні моделі життєвого циклу розробки об'єктно-орієнтованих програмних систем
6. Типи відношень, які використовуються в класових діаграмах
7. Роботи, що проводяться на етапі аналізу для життєвого циклу розробки програмних систем.
8. Способи відображення активних об'єктів в UML діаграмах.
9. Роботи, що виконуються при специфікації вимог до програмних систем.
10. Відношення спадкування в діаграмах класів.
11. Роботи, що виконуються на етапі системного аналізу
12. Відношення агрегації в класових діаграмах.
13. Роботи, що виконуються на етапі об'єктного аналізу.
14. Відношення асоціації в діаграмах класів та способи їх реалізації в об'єктно-орієнтованих мовах програмування
15. Роботи, що виконуються на етапі проектування.
16. Способи представлення багатоваріантних архітектур засобами UML.
17. Роботи, що виконуються на етапі проектування механізмів.
18. Подібності і відмінності автоматних моделей і діаграм Харела, що використовуються в діаграмах станів в UML.
19. Роботи, що виконуються на етапі детального проектування.
20. Параметризація класів в UML.
21. Роботи, що виконуються на етапі проектування архітектури.
22. Типи обмежень у вбудованих системах та системах реального часу.
23. Роботи, що виконуються на етапі кодування.
24. Використання UML-діаграм на різних етапах життєвого циклу розробки програмної системи.
25. Роботи, що виконуються на етапі тестування.
26. Мови специфікацій вимог до програмного забезпечення.
27. Діаграми прецедентів (UseCase діаграми) в UML.
28. Поняття мета класу.
29. Об'єктні діаграми (діаграми взаємодії) в UML.
30. Типи відношень в UseCase діаграмах.
31. Діаграми класів в UML.
32. Інструментальні засоби проектування, які використовують UML в якості вхідної мови.
33. Діаграми станів в UML.
34. Сценарії для UseCase діаграм та способи їх представлення.
35. Діаграми послідовностей в UML.
36. Відношення асоціації та його використання при генерації програм.
37. Діаграми розгортання в UML.
38. Способи задавання обмежень в діаграмах послідовностей в UML.
39. Діаграми компонентів в UML.
40. Класові діаграми та їх роль в діаграмах розгортання.
41. Проведення оцінки проекту на основі loc- і fr-метрик.
42. Аналіз чутливості програмного проекту на основі моделі COSOMO II

43. Принцип дії рефакторингу.
44. Метрики об'єктно-орієнтованих програмних систем.
45. Оптимізація коду, принцип дії.

12. Навчально-методичне забезпечення

Освітній процес з дисципліни «Конструювання програмного забезпечення» повністю і в достатній кількості забезпечений необхідною навчально-методичною літературою. Зокрема, викладачами кафедри підготовлені і видані такі роботи:

- Електронний аналог друкованого видання Конструювання програмного забезпечення: методичні рекомендації до виконання лабораторних робіт для здобувачів першого (бакалаврського) рівня вищої освіти спеціальності 121 «Інженерія програмного забезпечення Сертифікат №10е/20. Затверджено рішенням науково-методичної ради університету протокол №8 від 18.06.2023 р.

13. Матеріально-технічне та програмне забезпечення дисципліни (за потреби)

Інформаційна та комп'ютерна підтримка: ПК, планшет, смартфон або інший мобільний пристрій, проєктор. Програмне забезпечення: програми Microsoft Office або аналогічні, доступ до мережі Інтернет, робота з презентаціями.

Вивчення навчальної дисципліни не потребує використання спеціального програмного прикладного забезпечення, крім загальноновживаних програм і операційних систем.

14. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

ОСНОВНА:

1. Tejas Chopra, Dharendra Sinha. System Design Guide for Software Professionals: Build scalable solutions. Packt Publishing, The Limited. – 2024 – pp..384
2. Кунгурцев О.Б., Новікова Н.О. Конструювання програмного забезпечення. Об'єктно-орієнтований підхід. Видавництво Кондор. – 2024 – 228 стор.
3. Neal Ford, Mark Richards. Fundamentals of Software Architecture : An Engineering Approach. O'reilly Media, Incorporated – 2020 – pp..400
4. Joel Natty. Software Design Patterns for Scalable Applications Independently Published. – 2025 – pp..142
5. Piotr Sliż. Organizacja procesowo-projektowa. Istota, modelowanie, pomiar dojrzałości. Difin – 2021 – 292 str.
6. Ian Sommerville. Software Engineering, 10th edition. Published by Pearson . July 14th 2021 – 816 pp.
7. Роберт Мартін. Чистий код: Чистий код: створення, аналіз, рефакторинг. - Фабула. 2019 – 416 с.
8. Постіл. С.Д. UML. Уніфікована мова моделювання інформаційних систем: Навч. посіб., 2019. - 321 с.
9. Andrew Hunt, David Thomas. The Pragmatic Programmer: Your Journey To Mastery, 20th Anniversary Edition (2nd Edition). Print2print – 2020 – 352 стор.
10. Pinto J. Project Management: Achieving Competitive Advantage, 5th Edition. – 2019. – 592 p.
11. Маєвський Я. І., Праворська Н.І. Підвищення ефективності автоматизації масштабування мікросервісів у системі керування контейнеризованими застосунками KUBERNETES. - Вісник ХНУ, серія Технічні науки, N 5 , 2022
12. Seblewongel Esseynew Biale, Nuno Manuel Garcia, Dida Midekso, Nuno Pombo. Ethical Issues in Software Requirements Engineering. Software 2022, 1(1), 31- 52; <https://doi.org/10.3390/software1010003>

ДОПОМІЖНА:

13. Philippe Kruchten. The Rational Unified Process: An Introduction (5rd Edition) (Addison-wesley Object Technology Series). Addison-Wesley Professional; 5rd edition. 2018 – 407 p.
14. Shyam R Chidamber, Chris F Kemerer (Author), Sloan School of Management. A Metrics Suite for Object Oriented Design. Franklin Classics. - 2018 – 44 p.
15. D.Spinellis. Programcodeanalysis -AddisonWesley, 2004
16. ErichGamma, RichardHelm, RalphJohnson, andJohnVlissides. DesignPatterns: ElementsofReusable Object-Oriented Software.Addison-Wesley - 2009 - 417p.
17. Beck Kent. Extreme Programming Explained: EmbraceChange (The XP Series). Addison-WesleyProfessional. – 2004 - 224

18. Dave Nicolette. Software Development Metrics 1st Edition - Manning; 1st edition – 2015 – 192 p.
19. Eric J. Braude; Michael E. Bernstein. Software Engineering. Waveland Press, 2016 – 802 p.
20. Philipp Hohl, Jil Klünder, Arie van Bennekum, Ryan Lockard, James Gifford, Jürgen Münch, Michael Stupperich and Kurt Schneider. Back to the future: origins and directions of the “Agile Manifesto” – views of the originators Citation:Journal of Software Engineering Research and Development, 2018
21. Paulo Sérgio Medeiros dos Santos, Alessandro Caetano Beltrão, Bruno Pedraça de Souza and Guilherme Horta Travassos. On the benefits and challenges of using kanban in software engineering a structured synthesis study – Citation:Journal of Software Engineering Research and Development, 2018
22. Dijkstra, E. W. (Aug 1972). "The Humble Programmer". Communications of the ACM 15 (10): 859–866. doi:10.1145/355604.361591.
23. <http://www.cs.utexas.edu/~EWD/transcriptions/EWD03xx/EWD340.html>. (EWD340) PDF, 1972 ACM TuringAwardlecture
24. RussellGold, ThomasHammell, TomSnyder. TestDrivenDevelopment: A J2EE Example.- Apress, 2005.- 296 pages.
25. Pravorska N. METHOD OF APPLYING MACHINE LEARNING TO ENHANCE THE EFFICIENCY OF DEVOPS PROCESSES – November 2024. Herald of Khmelnytskyi National University Technical sciences 343(6(1)):454-463. DOI:10.31891/2307-5732-2024-343-6-68
26. Праворська Н.І., Грига С.І. МЕТОДИ РЕАЛІЗАЦІЇ МІКРОСЕРВІСНИХ АРХІТЕКТУР: ПЕРЕВАГИ ТА НЕДОЛІКИ, ВПРОВАДЖЕННЯ ТА ТЕСТУВАННЯ ПРИ РОЗРОБЦІ ПРОГРАМНИХ ЗАСТОСУНКІВ. Вісник Хмельницького національного університету №3, Т1. 2024 (335)- стор. 404-409

15. ІНФОРМАЦІЙНІ РЕСУРСИ

1. Модульне середовище для навчання. Доступ до ресурсу: <https://msn.khmnu.edu.ua/course/view.php?id=6688>
2. Електронна бібліотека університету . Доступ до ресурсу: <http://library.khmnu.edu.ua/>
3. Репозитарій ХНУ. Доступ до ресурсу: <http://elar.khmnu.edu.ua>