

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

ЗАТВЕРДЖУЮ

Декан факультету Інформаційних технологій

Тетяна Говорущенко

Підпис: _____

20 15 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ Основи командної розробки програмного забезпечення

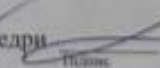
Призначення Робочої програми Для освітніх програм різних спеціальностей
Рівень вищої освіти Перший (бакалаврський)
Мова навчання Українська
Обсяг дисципліни, кредитів ЕКТС 8
Статус дисципліни Вибіркова
Факультет Інформаційних технологій
Кафедра Інженерії програмного забезпечення

Форма здобуття освіти	Курс	Семестр	Обсяг дисципліни		Кількість годин						Форма семестрового контролю
			Кредити ЕКТС	Години	Аудиторні заняття					Самостійна робота (в т.ч. ІРС)	Залік
					Разом	Лекції	Лабораторні роботи	Практичні заняття	Семінарські заняття		
Д	1	2	8	240	82	32	50			158	+

Робоча програма складена на основі освітніх програм підготовки бакалавра та стандарту вищої освіти спеціальності.

Робоча програма складена  к.пед. н., доцент, Наталія ПРАВОРСЬКА
(Підпис(и) автор(ів)) (Науковий ступінь, учасні знамен, ім'я, ПІРІМІЩЕ: автор(ів))

Схвалена на засіданні кафедри Інженерія програмного забезпечення

Протокол від 28 серпня 2025 р. № 1 Зав. кафедри  Госнід БЕДРАТЮК
(Підпис) (Ім'я, ПІРІМІЩЕ)

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

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

ОСНОВИ КОМАНДНОЇ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Дисципліна «Основи командної розробки програмного забезпечення» є вибірковою дисципліною, але необхідною для спеціалістів ІТ сфери для подальшого використання набутих знань та умінь при розробці програмного забезпечення. Вивчення навчальної дисципліни дозволяє здобувачам оволодіти знаннями в галузі командної розробки програмного забезпечення: розуміти основні етапи попередніх досліджень, визначення цілей продукту, який розробляється, зарубіжні та вітчизняні стандарти оформлення специфікації до ПЗ вимог, мови специфікації вимог, проводити збір вимог до програмного забезпечення застосовувати різні шаблони проектування, середовища розробки і архітектур в проектуванні ПЗ; розробляти, аналізувати і реалізовувати програмне забезпечення, використовуючи кілька різних технологій проміжного програмного забезпечення; оцінювати відповідність результатів проектування поставленим цілям; модифікувати проекти, використовуючи продумані підходи до управління змінами; використовувати методи зворотної інженерії (зворотне проектування) для відновлення дизайну програмного забезпечення.

Процес вивчення дисципліни спрямований на **формування таких професійних компетенцій, як здатність**: аналізувати предметні області, формувати, класифікувати вимоги до програмного забезпечення; розробляти і реалізовувати наукові та/або прикладні проекти у сфері інженерії програмного забезпечення; проектувати архітектуру програмного забезпечення, моделювати процеси функціонування окремих підсистем і модулів; розвивати і реалізовувати нові конкурентоспроможні ідеї в інженерії програмного забезпечення; розробляти, аналізувати та застосовувати специфікації, стандарти, правила і рекомендації в сфері інженерії програмного забезпечення; ефективно керувати фінансовими, людськими, технічними та іншими проектними ресурсами у сфері інженерії програмного забезпечення; 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; розробляти і координувати процеси, етапи та ітерації життєвого циклу програмного забезпечення на основі застосування сучасних моделей, методів та технологій розроблення програмного забезпечення; забезпечувати якість програмного забезпечення..

Зміст навчальної дисципліни. Основні підходи до командної розробки ПЗ, Microsoft Solutions Framework. Основні компоненти і принципи методології, управління проектами ризиками і готовністю в MSF, моделі процесів і команди MSF, програмні засоби управління і контролю командної розробки ПЗ, проектний менеджмент, управління комунікаціями в рамках команди, Сутність і роль архітектора програмного забезпечення, Роль розробника проекту, Механізми і способи тестування і налагодження проекту, особливості організації командної роботи.

Запланована навчальна діяльність: лекцій 32 год., лабораторних занять 50 год., самостійної роботи 158 год., разом 240 год.

Форми (методи) навчання: лекції (з використанням методів проблемного навчання і візуалізації), лабораторні заняття (з використанням комп'ютерних методів та засобів навчання), самостійна робота.

Форми оцінювання результатів навчання: усне опитування, захист лабораторних робіт

Вид семестрового контролю: залік – 5 семестр.

Навчальні ресурси:

1. Antti Niittyviita. Storytools of Testing: How To Get Your Voice Heard And Become Highly Valued Software Testing Professional. 2019 – 169 p.
2. Marijn Haverbeke. Eloquent JavaScript, 3rd Edition: A Modern Introduction to Programming. No Starch Press; 3rd edition (4 Dec. 2018). – 472 p.
3. Mariot Tsitoara. Beginning Git and GitHub: A Comprehensive Guide to Version Control, Project Management, and Teamwork for the New Developer. – 2019 – 308 p.
4. Pinto J. Project Management: Achieving Competitive Advantage, 5th Edition. – 2019. – 592 p. Бородкіна І., Бородкін Г.. Інженерія програмного забезпечення. – Центр навчальної літератури. – 2018.- 204 с.
5. Гайдаржи В.І., Изварін І.В. Бази даних в інформаційних системах. - Університет «Україна». – 2018.- 418 с.
6. Модульне середовище для навчання MOODLE. Доступ до ресурсу: <https://msn.khnu.km.ua>.

Викладач: кандидат педагогічних наук, доцент Праворська Н.І., асистент Дьоміна А.І.

3. Пояснювальна записка

Дисципліна “Основи командної розробки програмного забезпечення” є дисципліною вільного вибору здобувача рівня бакалавр. Метою вивчення навчальних дисциплін є отримання студентами знань та набуття практичних навичок у галузі конструювання програмного забезпечення. Для вирішення теоретичних і практичних завдань, що виникають при діяльності людини у різних галузях науки, техніки та виробництва великий ефект дає використання обчислювальної техніки при умові достатнього програмного забезпечення. Відповідно знання і вміння користуватись сучасним програмним забезпеченням стає нагальною потребою для фахівця з вищою освітою. Тому дисципліна “Основи командної розробки програмного забезпечення” у підготовці фахівців високої кваліфікації набуває особливо великого значення.

Пререквізити: введення в програмну інженерію, основи програмування, бази даних, алгоритми і структури даних, об’єктно-орієнтоване програмування, технології програмування на Java, іноземна мова (англійський).

Кореквізити: Web-програмування, прикладні інформаційні системи, конструювання програмного забезпечення.

Мета дисципліни – є забезпечення базової профілюючої підготовки за фахом, формування теоретичних знань та практичних навичок у галузі розробки програмного забезпечення, дати студентам базові знання про стандарти оформлення специфікації до ПЗ у відповідності до вимог. Здобуті у процесі вивчення дисципліни знання є базою для вивчення дисциплін професійно-орієнтованого циклу. Вивчення навчальної дисципліни дозволяє студентам оволодіти знаннями в галузі командної розробки програмного забезпечення: розуміти основні етапи попередніх досліджень, визначення цілей продукту, який розробляється, зарубіжні та вітчизняні стандарти оформлення специфікації до ПЗ вимог, мови специфікації вимог, проводити збір вимог до програмного забезпечення застосовувати різні шаблони проектування, середовища розробки і архітектур в проектуванні ПЗ; розробляти, аналізувати і реалізовувати програмне забезпечення, використовуючи кілька різних технологій проміжного програмного забезпечення; оцінювати відповідність результатів проектування поставленим цілям; модифікувати проекти, використовуючи продумані підходи до управління змінами; використовувати методи зворотної інженерії (зворотне проектування) для відновлення дизайну програмного забезпечення.

Предмет дисципліни. Предмет дисципліни “Основи командної розробки програмного забезпечення” охоплює теоретичні знання, задачі, функції та вимоги до програмного забезпечення. Основні підходи до командної розробки ПЗ, Microsoft Solutions Framework. Основні компоненти і принципи методології, управління проектами ризиками і готовністю в MSF, моделі процесів і команди MSF, програмні засоби управління і контролю командної розробки ПЗ, проектний менеджмент, управління комунікаціями в рамках команди, Сутність і роль архітектора програмного забезпечення, Роль розробника проекту, Механізми і способи тестування і налагодження проекту, особливості організації командної роботи.

Завдання дисципліни – отримання студентами знань, умінь і навичок в області застосування інформаційних систем, оволодіння дослідницькими навичками, уміння працювати самостійно, здатністю породжувати нові ідеї (креативністю), оволодіння основними методами, способами й засобами одержання, зберігання, переробки інформації з використанням комп’ютерної техніки, уміннями аналізувати й оцінювати зібрані дані.

Результати навчання. Студент, який успішно завершив вивчення дисципліни, має: досконало володіти професійною термінологією та основними поняттями ПЗ, розуміти основні етапи попередніх досліджень, визначення цілей продукту, що розробляється, зарубіжні та вітчизняні стандарти оформлення специфікації до ПЗ вимог, мови специфікації вимог, проводити збір вимог до програмного забезпечення застосовувати різні шаблони проектування, середовища розробки і архітектур в проектуванні ПЗ; розробляти, аналізувати і реалізовувати програмне забезпечення, використовуючи кілька різних технологій проміжного програмного забезпечення; використовувати адекватні метрики якості як засіб оцінки якості проектування, оцінювати відповідність результатів проектування поставленим цілям; модифікувати проекти, використовуючи продумані підходи до управління змінами; використовувати методи зворотної інженерії (зворотне проектування) для відновлення дизайну програмного забезпечення; мати навички командної розробки, погодження, оформлення і випуску всіх видів програмної документації.

4. Структура залікових кредитів дисципліни

“Основи командної розробки програмного забезпечення”

Теми	Лекції	Лаб. роб.	Самостійна робота студентів
Тема 1. Сучасні технології та інструментарій програмування. Призначення і роль новітніх інформаційних систем та технологій. Класифікація та можливості.	2	8	14
Тема 2. Початок роботи з проектом. Навички командної розробки проектів. Спеціалізовані середовища для розробки проектів.	12	8	36
Тема 3. Щоденний цикл роботи.	2	8	18
Тема 4. Розгалуження	2		18
Тема 5. Злиття версій	8	8	18
Тема 6. Конфлікти та їх вирішення	2	10	18
Тема 7. Блокування	2	8	18
Тема 8. Версії проекту, теги	2		18
Разом за 5-й семестр	32	50	158

5. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

“Основи командної розробки програмного забезпечення”

5.1. Зміст лекційного курсу*

№ лекції	Перелік тем лекцій, їх анотація	Кількість годин
Тема 1. Сучасні технології та інструментарій програмування. Призначення і роль новітніх інформаційних систем та технологій. Класифікація та можливості.		
1	Лекція 1. Основні поняття технологій програмування. Перелік сучасних технологій та можливих засобів, інструментів, програмного забезпечення для розв’язання поставлених задач. Нові тенденції в галузі інформаційних систем та технологій. Призначення та їх роль в розробці ПЗ. Класифікація та можливості. Сучасна класифікація технологій програмування та інформаційних систем. Література: [2; 9; 15].	2
Тема 2. Початок роботи з проектом. Навички командної розробки проектів. Спеціалізовані середовища для розробки проектів		
2	Лекція 2. Дії перед початком роботи з проектом. Спеціалізовані середовища розробки програмного забезпечення Eclips, CVS та Git. Вільне інтегроване середовище розробки модулів кросплатформених пропозицій – Eclips. Основні характеристики. Порядок установки. Переваги та недоліки. Принцип роботи. Вільна централізована система управління версіями, від компанії CollabNet – Subversion . Основні характеристики. Порядок установки. Переваги та недоліки. Принцип роботи. Література: [9-12]	2
3	Лекція 3. Спеціалізовані середовища розробки програмного забезпечення CVS та Git. Централізована система управління версіями – CVS (Concerrent Version Sistem, «Система одночасних версій»). Основні характеристики. Порядок установки. Переваги та недоліки. Принцип роботи. Розподілена система керування версіями – Git (Distributed revision control system). Основні характеристики. Порядок установки. Переваги та недоліки. Принцип роботи. Література: [9-12]	4
4	Лекція 4. Робота з універсальним інтегрованим середовищем NetBeans	2

	IDE. Універсальне інтегроване середовище NetBeans IDE. Основні характеристики. Порядок установки. Переваги та недоліки. Принцип роботи. Інтегроване середовище розробки програмного забезпечення для багатьох мовних програм IntelliJ IDEA. Основні характеристики. Порядок установки. Переваги та недоліки. Принцип роботи. Література: [9; 11].	
5	Лекція 5. Робота з пропрієтарним та інтегрованим середовищем розробки Embarcadero. Пропрієтарне середовище розробки на мові Java компанії Embarcadero, яке входить до складу Embarcadero RAD Studio – JBuilder. Основні характеристики. Порядок установки. Переваги та недоліки. Принцип роботи. Лінійка продуктів компанії Microsoft, яка включає в себе інтегроване середовище розробки програмного забезпечення та ряду інших інструментальних засобів – Microsoft Visual Studio. Основні характеристики. Порядок установки. Переваги та недоліки. Принцип роботи. Література: [11, 12].	4
Тема 3. Щоденний цикл роботи.		
6	Лекція 6. Етапи життєвого циклу програмних продуктів. Огляд існуючих типів життєвого циклу. Переваги та недоліки, правила роботи. Література: [9; 12-13]	2
Тема 4. Гілкування.		
7	Лекція 7. Опис послідовності розробки ПЗ. Визначення послідовності розробки, опис можливого гілкування. Література: [4; 6; 13]	2
Тема 5. Злиття версій.		
8	Лекція 8. Робота з Git. Злиття версій . Засади командної організації проектування ІС. Особливості, характеристики. Робота з програмним забезпеченням, яке забезпечує злиття версій на прикладі розробленого ПЗ. Початок роботи. Робочий процес, інтеграція з Github.	4
9	Лекція 9. Робота з Git. Злиття версій (продовження). Робочі директорії Working Directory. Складне злиття. Галуження та зливання. Злиття master з гілкою style. Розширення можливостей злиття. Перетворення, як альтернатива злиттю. Аналіз зроблених змін. Література: [16,18]	4
Тема 6. Конфлікти та їх вирішення.		
10	Лекція 10. Поняття конфліктів при роботі з системами керування версіями. Визначення ситуацій, які призводять до конфліктів. Шляхи та засоби уникнення. Вирішення конфлікту.. Література: [16, 17, 18].	2
Тема 7. Блокування.		
11	Лекція 11. Правила та засоби роботи з блокуванням в Git. Засоби та інструментарії блокування гілок. Блокування гілки за допомогою представлення «Гілки» Література: [16-18,20].	2
Тема 8. Версії проекту, теги.		
12	Лекція 12. Використання версій та тегів при роботі з Git Основні поняття. Характеристики, засоби для використання. Як отримати інформацію про версію, Повернутися до старої версії. Напівавтоматичне встановлення номера версії. Розподілений Git. Інші налаштування Git. Git та інші системи керування версіями. Підсумкове заняття. Література: [17, 19, 21].	2
Разом за 5-й семестр		32

5.2 Зміст лабораторних занять

№ з/п	Теми лабораторних занять	Години
-------	--------------------------	--------

1	Аналіз стану розвитку програмної інженерії в Україні. Початок розробки програмного забезпечення – стадія Технічне завдання.	8
2	Етапи життєвого циклу розробки програмного забезпечення. Специфікація вимог при розробці програмного забезпечення.	8
3	Методологія RUP (Rational Unified Process) при проєктуванні ПЗ.	8
4	Розробка планів проєктних робіт із створення програмного засобу. Сітковий графік та діаграма Ганта, побудова її в системі управління проєктами Microsoft Project.	8
5	Розробка або вибір алгоритму проєктування ПП. Виконання кодування. Робота з системою контролю версій Git у команді.	10
6	Налагодження та тестування програмного забезпечення.	8
Разом за 5-й семестр		50

5.3 Зміст самостійної роботи

Об'єм самостійної роботи з дисципліни “Основи командної розробки програмного забезпечення” становить 158 години. Вони включають опрацювання лекційного матеріалу, теоретичних і лабораторних завдань, підготовку до виконання лабораторних робіт, їх захисту і поточне тестування.

Самостійна робота здобувачів вищої освіти полягає у систематичному опрацюванні програмного матеріалу з відповідних джерел інформації, підготовці до лабораторних занять, виконанні індивідуальних завдань, тестування з теоретичного матеріалу тощо. Крім цього до послуг студентів сторінка навчальної дисципліни у Модульному середовищі для навчання, де розміщені Робоча програма дисципліни та необхідні документи з її навчально-методичного забезпечення.

Номер тижня	Назва теми	Години
1	Тема 1. Сучасні технології та інструментарій програмування. Призначення і роль новітніх інформаційних систем та технологій. Класифікація та можливості. Опрацювання лекційного матеріалу, підготовка та виконання лабораторної роботи №1. Література: [1; 3; 11].	6
2	Тема 1. Сучасні технології та інструментарій програмування. Призначення і роль новітніх інформаційних систем та технологій. Класифікація та можливості. Захист лабораторної роботи № 1. Література: [4; 7].	8
3	Тема2. Початок роботи з проєктом. Спеціалізовані середовища для розробки проєктів Опрацювання лекційного матеріалу, підготовка та виконання лабораторної роботи №2. Література: [5, 9; 12].	6
4	Тема2. Початок роботи з проєктом. Спеціалізовані середовища для розробки проєктів Опрацювання лекційного матеріалу, захист лабораторної роботи №2. Література: [5, 11].	6
5	Тема2. Початок роботи з проєктом. Спеціалізовані середовища для розробки проєктів Опрацювання лекційного матеріалу. Підготовка та виконання лабораторної роботи №3. Література: [7; 12].	8
6	Тема 2. Початок роботи з проєктом. Спеціалізовані середовища для розробки проєктів Опрацювання лекційного матеріалу, вконання лабораторної роботи №3. Література: [8, 12].	8
7	Тема 2. Початок роботи з проєктом. Спеціалізовані середовища для розробки проєктів Опрацювання лекційного матеріалу. Захист лабораторної роботи №3. Література: [9, 10].	8

	Підготовка до поточного контролю.	
8	Тема 3. Щоденний цикл роботи. Опрацювання лекційного матеріалу, підготовка та виконання лабораторної роботи №4. Література: [2-4; 8-9]	18
9	Тема 4. Розгалуження. Опрацювання лекційного матеріалу, захист лабораторної роботи №4. Література: [5,8]	18
10	Тема 5. Злиття версій. Опрацювання лекційного матеріалу, підготовка до виконання лабораторної роботи №5. Література: [7, 9]	8
11	Тема 5. Злиття версій. Опрацювання лекційного матеріалу, виконання лабораторної роботи №5. Література: [8-9]	10
12	Тема 6. Конфлікти та їх вирішення. Опрацювання лекційного матеріалу, виконання лабораторної роботи №5. Література: [5, 9].	8
13	Тема 6. Конфлікти та їх вирішення. Опрацювання лекційного матеріалу, виконання лабораторної роботи №5. Література: [9].	10
14	Тема 7. Блокування. Опрацювання лекційного матеріалу, підготовка до захисту лабораторної роботи №5. Література: [8-9].	18
15	Тема 8. Версії проекту, теги. Опрацювання лекційного матеріалу. Підготовка та виконання лабораторної роботи №6. Література: [6,8,9].	8
16	Тема 8. Версії проекту, теги. Опрацювання лекційного матеріалу. Захист лабораторної роботи №6. Література: [8-9]. Підготовка до іспиту.	10
Разом за 5-й семестр		158

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 практичних занять, а також у дні проведення контрольних заходів, встановлених робочою програмою і графіком освітнього процесу.

При цьому використовуються такі методи поточного контролю:

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

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 помилки .
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 несуттєві помилки .
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 суттєві помилки 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			Самостій на робота	Семест- ровий контроль	Разом
П'ятий семестр											
Лабораторні роботи №:						Опиту- вання	Тестовий контроль:		Доповідь	Залік	Сума балів
1	2	3	4	5	6	1-6	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		
18-30						18-30	12-20	6-10	6-10		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

Кількість балів	>12	12	16	20
-----------------	-----	----	----	----

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 пойдених модулів	0	1-2	1-3	1-4
Відсоток правильних відповідей	10%	25%	50%	100%
Отримана оцінка	-	3	4	5
Отримана кількість балів	-	6	8	10

Усне опитування

Усне опитування відбувається після кожної 2-ї лекції на лабораторній заняттях, згідно з питаннями представленими в кінці кожної лекції. Тобто на протязі всього курсу студенти мають 6 контрольних точок з опитуваннями. Орієнтовно питання варіативно узгоджуються з питаннями пунктів 11. *Питання для самоперевірки з дисципліни «Основи командної розробки програмного забезпечення»* та 12. *Питання до тесту з дисципліни «Основи командної розробки програмного забезпечення»*. Для оцінювання кожному студенту на кожній контрольній точці надається по 3 питання

Розподіл балів в залежності від наданих правильних відповідей на поставлені питання

Кількість вірних відповідей	0	1	1-2	1-3
Відсоток правильних відповідей	0	30%	60%	100%
Отримана оцінка	-	3	4	5

Доповідь

Форма оцінювання **Доповідь**, призначена для перевірки самостійної роботи студента на протязі вивчення дисципліни «Основи командної розробки програмного забезпечення». Кожний студент обирає теми (орієнтовно) для розгляду представлені в пункті 10. Теми, які поповнюються для підготовки доповіді. Доповідь представляє собою презентацію розглянутого питання з оформленням звіту згідно з вимогами ВНЗ. Сама форма оцінювання має містити наступні складові:

- оформлений звіт;
- презентація;
- доповідь на занятті.

Розподіл балів в залежності від виконання форми оцінювання Доповідь

Види роботи для форми оцінювання Доповідь	Оцінка	Кількість набраних балів
Наявність правильно оформленого звіту	3	6
Розроблена презентація	4	2
Доповідь	5	2

Підсумкова семестрова оцінка за інституційною шкалою і шкалою ЄКТС визначається в автоматизованому режимі після внесення викладачем результатів оцінювання у балах з усіх видів

навчальної роботи до електронного журналу. Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС наведені нижче у таблиці «Співвідношення».

Семестровий залік виставляється на останньому занятті за умови якщо загальна сума балів, яку накопичив здобувач з дисципліни (іншого освітнього компонента) за результатами **поточного** контролю, знаходиться у межах від 60 до 100 балів. При цьому за інституційною шкалою ставиться оцінка «**зараховано**», а за шкалою ЄКТС – буквене позначення оцінки, що відповідає набраній студентом сумі балів відповідно до таблиці Співвідношення. Присутність здобувача у цьому випадку не є обов'язковою.

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		
D	66-72		Задовільно/Satisfactory – Наявні мінімально достатні для подальшого навчання та/або професійної діяльності за фахом результати навчання з навчальної дисципліни
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. Теми, які поповнюються для підготовки доповіді

1. Життєвий цикл розробки програмного забезпечення (SDLC)
2. Технологія розробки ПЗ
3. Аналіз та проектування ПЗ
4. Реалізація та тестування ПЗ
5. Впровадження та супровід ПЗ
6. Методології та підходи. Waterfall (Каскадна модель).
7. Методології та підходи. Послідовне виконання етапів розробки.
8. Agile. Гнучка методологія, що фокусується на ітеративної розробки та частого зворотного зв'язку.
9. Scrum. Фреймворк, що входить до сімейства Agile, з чітко визначеними ролями та церемоніями.
10. Lean. Методологія, спрямована на мінімізацію втрат та підвищення ефективності.
11. Командна робота та управління. Управління проектом.
12. Командна робота та управління. Комунікація у команді.
13. Командна робота та управління. Управління вимогами.
14. Інструменти та технології для розробки ПЗ
15. Системи контролю версії.
16. Task-трекери та системи управління завданнями. ПЗ для відстеження прогресу виконання завдань та багів (наприклад, Jira).
17. Інтегровані середовища розробки (IDE):
18. Як працювати з кількома віддаленими репозиторіями?
19. Як вирішувати конфлікти під час злиття?

20. Що таке gitignore і як його використовувати?
21. Як працювати з тегами?
22. Як використовувати Git Hooks?
23. Які існують стратегії розгалуження (наприклад, Gitflow)?
24. Як використовувати Git для спільної роботи над проектом?
25. Як використовувати Git для деплою?
26. Як використовувати Git для тестування?
27. В яких ситуаціях Git може бути корисним, а в яких ні?
28. Які є альтернативи Git?
29. Які переваги та недоліки розподілених систем контролю версій у порівнянні з централізованими?
30. Як Git впливає процес розробки ПЗ?
31. Які найкращі практики використання Git у команді?

11. Питання для самоперевірки з дисципліни “Основи командної розробки програмного забезпечення”

1. Поняття системи керування версіями.
2. Що розуміється під терміном «керування версіями»?
3. Описати структуру системи керування версіями.
4. Описати класифікацію системи керування версіями.
5. Описати призначення систем керування версіями.
6. Перерахувати рівні управління в системах керування версіями.
7. Навести приклади найбільш відомих систем керування версіями.
8. Дати визначення і перерахувати системи керування версіями.
9. Дати визначення задачі можливостям застосування систем керування версіями.
10. Визначення порядку роботи у системах керування версіями.
11. Назвати особливості щоденного циклу роботи.
12. Визначити оновлення робочої копії.
13. Визначити модифікацію проекту.
14. Як проходить визначення фіксації змін?
15. Дати визначення розгалуження проекту.
16. Дати визначення злиттю версій.
17. Дати визначення конфлікту та шляхи його вирішення.
18. Дати визначення блокуванню змін
19. Дати визначення версії проекту та тегам.
20. Назвати базові принципи розробки ПЗ у системах керування версіями.
21. Перерахувати базові принципи розробки ПЗ у розподілених системах керування версіями.
22. Дати визначення терміну та пояснити застосування команди branch.
23. Дати визначення терміну та пояснити застосування команди changeset.
24. Дати визначення терміну та пояснити застосування команди activity.
25. Дати визначення терміну та пояснити застосування команди check-in.
26. Дати визначення терміну та пояснити застосування команди commit.
27. Дати визначення терміну та пояснити застосування команди submit.
28. Дати визначення терміну та пояснити застосування команди check-out.
29. Дати визначення терміну та пояснити застосування команди clone.
30. Дати визначення терміну та пояснити застосування команди conflict.
31. Дати визначення терміну та пояснити застосування команди head.
32. Дати визначення терміну та пояснити застосування команди merge.
33. Дати визначення терміну та пояснити застосування команди integration.
34. Дати визначення терміну та пояснити застосування команди rebase.
35. Дати визначення терміну та пояснити застосування команди repository.
36. Дати визначення терміну та пояснити застосування команди depot.
37. Дати визначення терміну та пояснити застосування команди revision.
38. Дати визначення терміну та пояснити застосування команди shelving.
39. Дати визначення терміну та пояснити застосування команди tag.
40. Дати визначення терміну та пояснити застосування команди label.
41. Дати визначення терміну та пояснити застосування команди trunk.
42. Дати визначення терміну та пояснити застосування команди mainline.
43. Дати визначення терміну та пояснити застосування команди master.
44. Дати визначення терміну та пояснити застосування команди update.

45. Дати визначення терміну та пояснити застосування команди sync.
46. Дати визначення терміну та пояснити застосування команди working copy.
47. Дати характеристику й описати принципи роботи універсального інтегрованого середовища NetBeans IDE.
48. Дати характеристику й описати принципи роботи інтегрованого середовища розробки програмного забезпечення для багатьох мовних програм IntelliJ IDEA.
49. Дати характеристику й описати принципи роботи пропрієтарного середовища розробки на мові Java компанії Embarcadero, яке входить до складу Embarcadero RAD Studio – JBuilder.
50. Дати характеристику й описати принципи роботи інтегрованого середовища розробки програмного забезпечення та ряду інших інструментальних засобів – Microsoft Visual Studioю
51. Дати характеристику й описати принципи роботи вільного інтегрованого середовища розробки модулів кросплатформених пропозицій – Eclipse.
52. Дати характеристику й описати принципи роботи вільної централізованої системи управління версіями, від компанії CollabNet – Subversion.
53. Дати характеристику й описати принципи роботи централізованої системи управління версіями – CVS (Concurrent Version System, «Система одночасних версій»).
54. Дати характеристику й описати принципи роботи розподіленої системи керування версіями – Git (Distributed revision control system).

12. Питання до тесту з дисципліни

“Основи командної розробки програмного забезпечення”

1. Основні задачі проектування ІС полягають в:
2. Проектування ІС охоплює три основні області:
3. SWEBOK – це
4. Яка галузь знань, записана в SWEBOK
5. Software design – це
6. Типи нотацій
7. Назвати рівні моделі оцінки зрілості технологічних процесів в установах програмування (CMM)
8. Характеристика початкового рівня моделі оцінки зрілості технологічних процесів:
9. Характеристика повторюваного рівня моделі оцінки зрілості технологічних процесів
10. Характеристика визначеного рівня моделі оцінки зрілості технологічних процесів
11. Характеристика керованого рівня моделі оцінки зрілості технологічних процесів
12. Характеристика оптимізованого рівня моделі оцінки зрілості технологічних процесів
13. Життєвий цикл
14. Базові групи всіх процесів ЖЦ
15. Допоміжні процеси при створенні ПП:
16. Яка з моделей життєвого циклу є прямолінійною і найбільш простою для застосування
17. Яка з моделей життєвого циклу відповідає тестуванню та порівнянню результатів фаст-тестування та проектування
18. Яка з моделей життєвого циклу відповідає, щоб зв'язок між замовником забезпечується більш ефективно на стадіях оцінки прототипу
19. Яка з моделей життєвого циклу дає можливість значно зменшити цикл розробки програмного продукту завдяки групам в 3-7 висококваліфікованих спеціалістів, що здійснюють тісний взаємозв'язок з замовником
20. Яка з моделей життєвого циклу дає можливість внесення змін в процесі розробки ПП і є дуже швидкою
21. Яка з моделей життєвого циклу забезпечує можливість охоплення каскадної моделі розділяє фази розробки на більш дрібні частини, що забезпечує гнучкість при проектуванні
22. Вибрати з переліку переваги каскадної моделі
23. Вибрати з переліку переваги V-подібної моделі
24. Вибрати з переліку переваги моделі прототипу
25. Вибрати з переліку переваги RAD– моделі
26. Вибрати з переліку переваги моделі багатьох проходів
27. Вибрати з переліку переваги спіральної моделі
28. Вибрати з переліку недоліки каскадної моделі

29. Вибрати з переліку недоліки V-подібної моделі
30. Вибрати з переліку переваги RAD– моделі
31. Вибрати з переліку недоліки моделі багатьох проходів
32. Вибрати з переліку недоліки моделі прототипів
33. Етапи формування проектних груп
34. Junior Developer – це
35. Middle dev (розробник) – це
36. Senior developer (провідний розробник-програміст) – це
37. Team Lead (Тимлід) – це
38. Вказати функції (знання) Junior Developer
39. Вказати функції (знання) Middle dev (розробника):
40. Вказати функції (знання) Senior developer (провідний розробника-програміста)
41. Вказати функції (знання) Team Lead (Тимлід)
42. Система управління версіями – це
43. Pre-alpha(pa) – це
44. Alpha(a) – це
45. Beta(b) – це
46. ReleaseCandidate(rc) – це
47. Release to marketing (rtm) – це
48. Generalavailability(ga) – це
49. VCS для розробників забезпечує
50. За структурою VCS поділяється на
51. За характером роботи з об'єктами VCS поділяється на
52. За типом об'єкта VCS поділяється на
53. Репозиторій (repository, depot) – це
54. Гілка (branch) – це
55. Вибрати перевагу централізованих систем CVS,Subversion
56. Вибрати недолік централізованих систем CVS,Subversion
57. Вибрати серед перерахованих розподілену систему контролю версій:
58. VCS реалізує такі завдання командної роботи
59. В Git файли можуть знаходитись в одному з трьох станів
60. Місце, де Git зберігає метадані та базу даних об'єктів вашого проекту, найбільш важлива частина Git, тому, що вона копіюється, коли проодиться клонування репозиторію – це
61. Зроблена копія з бази певної версії проекту, де файли розміщуються в стиснутій базі даних у каталозі Git та розміщуються на диску для того, щоб їх переглядали та редагували – це
62. Звичайний файл, зазвичай який зберігається в каталозі Git, який вміщує інформацію про те, що повинно увійти до наступного комміту – це
63. Які з гілок Git зайві
64. Оновлення робочої копії – це коли
65. Фіксація змін – це коли
66. Злиття гілок – це коли
67. Конфлікт – це
68. Конфлікт може виникати в разі
69. Яка операція використовується для розв'язання конфлікту
70. Яка з стадій розробки ПП займає

12. Навчально-методичне забезпечення

Освітній процес з дисципліни «Основи командної розробки програмного забезпечення» повністю і в достатній кількості забезпечений і представлений у модальному середовищі ХНУ <https://msn.khmnu.edu.ua/course/view.php?id=5177>

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

Інформаційна та комп'ютерна підтримка: ПК, планшет, смартфон або інший мобільний пристрій, проектор. 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. Antti Niittyviita. Storytools of Testing: How To Get Your Voice Heard And Become Highly Valued Software Testing Professional. 2019 – 169 p.
2. Marijn Haverbeke. Eloquent JavaScript, 3rd Edition: A Modern Introduction to Programming. No Starch Press; 3rd edition (4 Dec. 2018). – 472 p.
3. Robin Nixon. Learning PHP, MySQL & JavaScript: A Step-by-Step Guide to Creating Dynamic Websites (Learning PHP, MYSQL, Javascript, CSS & HTML5) 6th Edition. O'Reilly Media. – 2021 – 823 p.
4. Pinto J. Project Management: Achieving Competitive Advantage, 5th Edition. – 2019. – 592 p.
5. Mariot Tsitoara. Beginning Git and GitHub: A Comprehensive Guide to Version Control, Project Management, and Teamwork for the New Developer. – 2019 – 308 p.
6. Tracy Osborn. Really Friendly Git Intro: Learn the basics of Git, the version control system for programming. Independently published. 2019. – 25 p.
7. Jameson Garner. GIT: The Ultimate Guide for Beginners: Learn Git Version Control.-Independently published. 2020. – 55 p.
8. François Dupire . Git Essentials: Developer's Guide to Git. Independently published. 2021. – 187 p.
9. Raju Gandhi. Head First Git: A Learner's Guide to Understanding Git from the Inside Out. O'Reilly Media; 1st edition. 2022 – 506 p.
10. Ralph Kliem. Developing a Network Diagram (The Fundamentals Book 402). Kindle Edition. – LeanPM, LLC – 2016 – 9 p.
11. The Art of Service - Microsoft Project Publishing. Microsoft Project A Complete Guide – 2021.. – 2020 – 311 p.
12. George Koelsch. Requirements Writing for System Engineering. – Apress; 1st ed. Edition – 424 p.
13. Gerardus Blokdyk. Rational Unified Process A Complete Guide. – 5STARCooks – 2020 – 308 p.
14. Гайдаржи В.І., Изварін І.В. Базы даних в інформаційних системах. - Університет «Україна». – 2018.- 418 с.

ДОПОМІЖНА:

1. ДСТУ 3278-95 Система розроблення та поставлення продукції на виробництво. Основні терміни та визначення.
2. ДСТУ 3321:2003 Система конструкторської документації. Терміни та визначення основних понять.
3. ДСТУ 3973-2000 Система розроблення та поставлення продукції на виробництво. Правила виконання науково-дослідних робіт. Загальні положення.
4. ДСТУ 3974–2000 Система розроблення та поставлення продукції на виробництво. Правила виконання дослідно-конструкторських робіт. Загальні положення.
5. ДСТУ ISO 11442-3:2004 Документація на технічну продукцію. Автоматизоване оброблення технічної інформації. Частина 3. Стадії процесу проектування продукції
6. <https://uk.myservername.com/top-40-static-code-analysis-tools>
7. <https://silo.tips/download/requirements-analysis-through-viewpoints-oriented-requirements-model-vord>
8. https://edisciplinas.usp.br/pluginfile.php/5043740/mod_resource/content/1/JME2009_p131_140.pdf
9. http://online.budstandart.com/ua/catalog/doc-page?id_doc=67052
10. http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=50970
11. http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=69133
12. http://online.budstandart.com/ua/catalog/doc-page.html?id_doc=69141
13. Microsoft Visual Studio Licensing. 2019. - Microsoft Corporation – 34 p.
14. 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
15. 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

16. Форкун Ю., Мартинюк В., Праворська Н., Лучицький О. Метрика диференційованої цикломатичної складності аналізу програмного коду з використанням систем керування базами даних. Measuring and computing devices in technological processes. 2023. №3. С. 100–105. DOI: DOI: 10.31891/2219-9365-2023-75-11

15. ІНФОРМАЦІЙНІ РЕСУРСИ

1. Модульне середовище для навчання. Доступ до ресурсу: <https://msn.khmnu.edu.ua/course/view.php?id=5177>
2. Електронна бібліотека університету . Доступ до ресурсу: <http://library.khmnu.edu.ua/>
3. Репозитарій ХНУ. Доступ до ресурсу: <http://elar.khmnu.edu.ua>