МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЗАХІДНОУКРАЇНСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

ЗАТВЕРДЖУЮ:
Декан факультету комп'ютерних інформаційних технологій
_____ Ігор ЯКИМЕНКО
"30" _____ 2024 р.

ЗАТВЕРДЖУЮ:
Проректор з науково-педагогічної роботи
_____ Віктор ОСТРОВЕРХОВ
"__" _____ 2024 р.

РОБОЧА ПРОГРАМА

з дисципліни «**Технологія розробки програмного забезпечення**»

ступінь вищої освіти – **бакалавр**

галузь знань – **12 "Інформаційні технології"**

спеціальність – **122 „Комп'ютерні науки"**

освітньо-професійна програма – **„ Штучний інтелект "**

Кафедра інформаційно-обчислювальних систем і управління

Форма навчання	Курс	Семестр	Лекції (год.)	Лабораторні заняття (год.)	ІРС (год.)	Тренінг (год.)	Самост. робота студ. (год.)	Разом (год.)	Екз. (сем.)
Денна	3	6	46	44	5	12	73	180	6

30.08.2024

Тернопіль – ЗУНУ
2024

Робоча програма складена на основі освітньо–професійної програми «Штучний інтелект» підготовки бакалавра галузі знань 12 “Інформаційні технології” спеціальності 122 «Комп’ютерні науки», затвердженої Вченою радою ЗУНУ (протокол № 9 від 15 червня 2022 р.) зі змінами відповідно до рішення Вченої ради ЗУНУ (протокол №11 від 26.06.2024р.).

Робочу програму склав викладач кафедри інформаційно-обчислювальних систем і управління Христина ЛІП'ЯНІНА-ГОНЧАРЕНКО

Робоча програма затверджена на засіданні кафедри інформаційно-обчислювальних систем і управління, протокол № 1 від 27 серпня 2024 р.

В.о. завідувача кафедри Надія ВАСИЛЬКІВ

Розглянуто та схвалено групою забезпечення спеціальності „Комп’ютерні науки”, протокол № 1 від 30 серпня 2024 р.

Голова групи
забезпечення спеціальності,
д-р техн. наук, професор Мирослав КОМАР

Гарант освітньо-професійної
програми "Штучний інтелект ",
к.т.н, доцент Василь КОВАЛЬ

СТРУКТУРА РОБОЧОЇ ПРОГРАМИ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ " ТЕХНОЛОГІЯ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ "

1. Опис дисципліни "Технологія розробки програмного забезпечення "

Дисципліна «Технологія розробки програмного забезпечення»	**Галузь знань, спеціальність, СВО**	**Характеристика навчальної дисципліни**
Кількість кредитів – 6	Галузь знань – 12 “Інформаційні технології”	**Статус дисципліни:** обов’язкова дисципліна циклу професійної підготовки **Мова навчання:** Українська
Кількість залікових модулів – 5	Спеціальність – 122 «Комп’ютерні науки»	Рік підготовки: 3 Семестр: *Денна* – 6
Кількість змістових модулів – 2	освітньо-професійна програма «Штучний інтелект»	Лекції: *Денна* –46 год. Лабораторні заняття: *Денна* – 44 год.
Загальна кількість годин – 180	Ступінь вищої освіти – бакалавр	Самостійна робота: *Денна* – 73 год., Тренінг: *Денна* – 12 год. Індивідуальна робота: *Денна* – 5 год.
Тижневих годин – 12, з них аудиторних – 6 год.		Вид підсумкового контролю – екзамен

2. Мета і завдання дисципліни "ТЕХНОЛОГІЯ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ"

2.1. Мета вивчення дисципліни

Мета дисципліни – є формування системи теоретичних знань і набуття практичних умінь і навичок з питань теоретичних основ та використання сучасних методології та технологій розроблення, тестування та експлуатації програмних продуктів, та для моделювання і проектування інформаційних систем, документування та маркетингу програмного забезпечення.

2.2. Завдання вивчення дисципліни

Завдання навчальної дисципліни «Технологія розробки програмного забезпечення» – вивчення понять, принципів, методології та технологій створення програмних продуктів як сукупності процесів розроблення програмних систем на засадах життєвого циклу (ЖЦ) програмного забезпечення інформаційних систем; вивчення призначення засобів об'єктного підходу до проектування програмного забезпечення інформаційних систем; вивчення стандартів програмної інженерії; вивчення методів створення вимог під час розроблення програмних продуктів; вивчення методів та засобів тестування програмне забезпечення сучасними засобами тестування програмних продуктів; вивчення стандартів якості під час створення програмного забезпечення інформаційних систем; вивчення інструментальних засобів програмної інженерії та їх практичне застосування під час проектування програмне забезпечення інформаційних систем.

2.3. Найменування та опис компетентностей, формування яких забезпечує вивчення дисципліни:

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК9. Здатність працювати в команді.

ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.

СК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об’єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

СК10. Здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника.

СК13. Здатність до розробки мережевого програмного забезпечення, що функціонує на основі різних топологій структурованих кабельних систем, використовує комп’ютерні системи і мережі передачі даних та аналізує якість роботи комп’ютерних мереж.

2.4. Передумови для вивчення дисципліни

Вивчення курсу „Технологія розробки програмного забезпечення” передбачає наявність систематичних та ґрунтовних знань із дисципліни «Розробка клієнт-серверних застосувань».

2.5. Результати навчання

ПР9. Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв’язання задач в галузі комп’ютерних наук.

ПР11. Володіти навичками управління життєвим циклом програмного

забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог і обмежень замовника, вміти розробляти проектну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт).

3. Програма навчальної дисципліни «ТЕХНОЛОГІЯ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»

Змістовий модуль 1 - Загальні особливості розробки програмного забезпечення

Тема 1. Базові поняття створення програмних продуктів

Поняття моделі життєвого циклу програмного забезпечення та інформаційної системи. Основні етапи життєвого циклу інформаційної системи – аналіз, розроблення вимог, проектування, тестування, ввід у експлуатацію, супровід. Причини виникнення технології програмної інженерії. Структура життєвого циклу за міжнародним стандартом ISO/IEC 12207. Концепція інформаційної безпеки та принципи безпечного проєктування програмного забезпечення.

Тема 2. Класифікація та сутність процесів розроблення програмного забезпечення

Принципи і методи проектування програмного забезпечення. Методи програмної інженерії. Рівні управління вимогами. Особливості застосування та формулювання вимог.

Тема 3. Моделі життєвого циклу ПЗ.

Технології розроблення програмного забезпечення. Програмний продукт. Життєвий цикл програмного забезпечення. Моделі життєвого циклу. Стратегії розробки ПЗ. Водоспадна модель (Waterfall model) ЖЦ. Інкрементна (incremental) модель. Макетування. Спіральна модель. Компонентно-орієнтована модель. RAD-модель. V- модель (V-Model)

Тема 4. Гнучкі методології.

Методологія Scrum. Важкі та полегшені процеси. Agile-маніфест. Методологія Kanban. Методологія XP (Extreme Programming). Методологія Scrum. Порівняння Agile-методологій

Тема 5. Інженерія вимог до ПЗ.

Методики формування вимог. Вимоги до програмного забезпечення. Категорії вимог. Етапи інженерії вимог. Типи вимог. Формування вимог. Діаграма прецедентів. Аналіз вимог (класичний підхід). Характеристики якісних вимог. Опис вимог у вигляді текстових сценаріїв. Формування та аналіз вимог в процесі Scrum. Історія користувача (user story). Критерії якісної історії користувача. Ієрархія історій. Критерії якісної задачі. Карта історій.

Тема 6. Аналіз вимог до ПЗ.

Опис детальних вимог за допомогою діаграм діяльності та станів UML. Мова моделювання UML. Використання UML для аналізу вимог. Режими використання. Режим ескізу. Режим проектування. Режим мови програмування. Типи діаграм UML 2.0. Діаграма діяльності. Діаграма станів.

Тема 7. Проектування програмного забезпечення.

Діаграма класів та об'єктів UML. Проектування програмного забезпечення. Використання нотації UML для проектування ПЗ. Діаграма класів: клас, ім'я, видимість полів та атрибутів, атрибути, операції, відношення між класами, генерування коду. Діаграма об'єктів.

Тема 8. Сучасні парадигми програмування.

Парадигма функціонального програмування. Парадигма логічного програмування. Парадигма об'єктно-орієнтованого програмування та декларативні засоби програмування.

Тема 9. Експертні системи у комп'ютерно-інтегрованих системах.

Загальні відомості про експертні системи. Означення та структура експертних систем. Інженерія знань, системи знань. Специфікація та моделі прийняття рішень.

Тема 10. Якість ПЗ.

Якість ПЗ. Види якості. Модель якості програмних систем. Модель Мак-Кола. Модель якості ISO/IEC 9126. Стандартні показники якості. Метрики якості.

Тема 11. Проектування UI.

Місце прототипу в життєвому циклі ПЗ. Стандарти проектування людино-машинного інтерфейсу. Засоби створення прототипів.

Тема 12. Об’єктні моделі та реляційні бази даних.

Архітектурні рішення. Функціональні проблеми. Зчитування даних. Взаємне відображення об’єктів та реляційних структур. Реалізація відображення. Наслідування з одною таблицею (Single Table Inheritance). Наслідування з таблицями для кожного класу (Class Table Inheritance). Наслідування з таблицями для кожного конкретного класу (Concrette Table Inheritance).

Тема 13. Модульне тестування.

Призначення. Переваги. Характеристики якісного теста. Тестові двійники: об'єкт-заглушка (Dummy Object), тестова заглушка (Test Stub), тестовий агент (Test Spy), імітація (Fake), підставний об'єкт (Mock). TDD. Фреймворки тестування. Підтримка модульного тестування на рівні мови.

Змістовий модуль 2 – Керування конфігурацією, архітектура тестування та супровід ПЗ

Тема 14. Системи контролю версій.

Основи Git. Про контроль версій. Особливості GIT: принципи збереження даних, стани файлу в робочому каталозі, процеси роботи з гілками, робочий процес одного розробника, робочий процес взаємодії декількох розробників, хостинг репозиторіїв, GitHub.

Тема 15. Безперервна інтеграція та розгортання.

Концепція безперервної інтеграції. Автоматизація складання. Гаки (hooks) в системах контролю версій. Інструменти складання. Засоби безперервної інтеграції. Концепція безперервного розгортання. Засоби безперервного розгортання.

Тема 16. Принципи проектування.

Шаблони DI. Зв’язність та зв’язаність. Інверсія залежності (Dependency Inversion). Шаблони DI: впровадження в конструктор, впровадження у властивість, впровадження в метод, навколишній контекст (Ambient Context). Антишаблони Dependency Injection: Диктатор (Control Freak), «Гібридне впровадження» (Bastard Injection), Обмежене конструювання» (Constrained Construction), Локатор сервісів (Service Locator). Принципи SOLID: принцип єдиної відповідальності (SRP), принцип відкритості / закритості (Open / Closed Principle), принцип підстановки Лісков (Liskov Substitution Principle), принцип поділу інтерфейсів (Interface Segregation Principle), принцип інверсії залежностей (Dependency Inversion Principle, DIP). Принцип KISS. Принцип YAGNI. Принцип DRY. Принцип «Говори, а не питай» (TellDon't-Ask).

Тема 17. Архітектурні стилі.

Огляд. Поєднання архітектурних стилів. Архітектура клієнт/сервер. Компонентна архітектура. Проектування на основі предметної області. Багатошарова архітектура (N-layer architecture). Гексагональна архітектура (Hexagonal Architecture). N-рівнева / 3-рівнева архітектура (N-tier architecture). Об’єктно-орієнтована архітектура. Сервісно-орієнтована архітектура (Service-Oriented Architecture). Мікро-сервісна архітектура.

Тема 18. Багатошарова архітектура.

Концепція розшарування. Архітектура базових типів застосунків. Мобільні

застосунки. Веб-застосунок. Особливості типових шарів. Рівень представлення. Шаблони реалізації бізнес логіки. Шаблони реалізації шару доступу до даних. Шар служб. Наскрізна функціональність. Чиста архітектура.

Тема 19. Породжуючі шаблони проектування.

Factory Method. Абстрактна фабрика (Abstract Factory). Одинак (Singleton). Прототип (Prototype). Будівельник (Builder).

Тема 20. Структурні шаблони проектування.

Декоратор (Decorator). Адаптер (Adapter). Фасад (Facade). Компоновщик (Composite). Заступник (Proxy). Міст (Bridge). Пристосуванець (Flyweight).

Тема 21. Поведінкові шаблони проектування.

Стратегія (Strategy). Спостерігач (Observer). Команда (Command). Шаблонний метод (Template Method). Ітератор (Iterator). Стан (State). Ланцюжок Обов'язків (Chain of responsibility). Інтерпретатор (Interpreter). Посередник (Mediator). Хранитель (Memento). Відвідувач (Visitor).

Тема 22. Безпечне проєктування програмного забезпечення

Сутність та методи розроблення безпечного програмного забезпечення. Аналіз та оцінка вразливостей інформаційних систем. ідентифікація вразливостей розроблення програмного забезпечення. Управління розробленням безпечних програмних систем.

4. Структура залікового кредиту з дисципліни «Технологія розробки програмного забезпечення»

Денна форма навчання

Тема	*Кількість годин*					
	Лекції	Лабораторні заняття	Індивідуальна робота	Тренінг	Самостійна робота	Контрольні заходи
Змістовий модуль 1 - Загальні особливості розробки програмного забезпечення						
Тема 1. Базові поняття створення програмних продуктів	2	-	-	6	3	Опитування під час заняття
Тема 2. Класифікація та сутність процесів розроблення програмного забезпечення	2	6	-		3	Опитування під час заняття
Тема 3. Моделі життєвого циклу ПЗ.	4	-	1		4	Опитування під час заняття
Тема 4. Гнучкі методології.	2	7	-		4	Опитування під час заняття
Тема 5. Інженерія вимог до ПЗ.	2	-	-		3	Опитування під час заняття
Тема 6. Аналіз вимог до ПЗ.	2	6	-		4	Опитування під час заняття
Тема 7. Проектування програмного забезпечення.	2	7	1		3	Опитування під час заняття
Тема 8. Сучасні парадигми програмування.	2	6	1		4	Опитування під час заняття

Тема 9. Експертні системи у комп'ютерно-інтегрованих системах.	2	-	-		4	Опитування під час заняття
Тема 10. Якість ПЗ.	2	4	1		3	Опитування під час заняття
Тема 11. Проектування UI.	2	-	-		3	Опитування під час заняття
Тема 12. Об’єктні моделі та реляційні бази даних.	2	-	-		4	Опитування під час заняття
Тема 13. Модульне тестування.	2	4	-		3	Опитування під час заняття
Змістовий модуль 2 – Керування конфігурацією, архітектура тестування та супровід ПЗ						
Тема 14. Системи контролю версій	2	4	-	6	3	Опитування під час заняття
Тема 15. Безперервна інтеграція та розгортання	2	-	-		3	Опитування під час заняття
Тема 16. Принципи проектування	2	-	-		3	Опитування під час заняття
Тема 17. Архітектурні стилі	2	-	1		3	Опитування під час заняття
Тема 18. Багатошарова архітектура	2	-	-		4	Опитування під час заняття
Тема 19. Породжуючі шаблони проектування	2	-	-		3	Опитування під час заняття
Тема 20. Структурні шаблони проектування	2	-	-		3	Опитування під час заняття
Тема 21. Поведінкові шаблони проектування	2	-	-		3	Опитування під час заняття
Тема 22. Безпечне проєктування програмного забезпечення	2	-	-		3	Опитування під час заняття
Разом	46	44	5	12	73	

5. Тематика лабораторних занять

Лабораторна робота № 1.
Тема: Вибір програмного проекту, формування команди, підготовка статуту.

Лабораторна робота № 2
Тема: План проекту.

Лабораторна робота № 3
Тема: Специфікація вимог до програмної системи.

Лабораторна робота № 4.
Тема: Стандарт UML.

Лабораторна робота № 5.
Тема: UML.

Лабораторна робота № 6.
Тема: UI/UX дизайн.

Лабораторна робота №7
Тема: Тестування програмної системи

Лабораторна робота №8
Тема: Система контролю версій Git. GitHub.

6. Самостійна робота

Студенти проводять дослідження на обрану з поданого нижче переліку тему та готують результати дослідження у вигляді реферату. Студенти можуть самостійно запропонувати тему дослідження, відповідно до власних вподобань та в рамках даного предмету і погодити її з викладачем.

Перелік тем:

1. Базові поняття та види програмного забезпечення.
2. Розроблення програмного забезпечення.
3. Поняття технології програмування як процесу.
4. Поняття життєвого циклу програмного забезпечення.
5. Каскадна модель (waterflow model).
6. Ітеративна модель (Iterative and incremental development).
7. Спіральна модель.
8. Розроблення ПЗ як проектна діяльність.
9. Основні форми планів робіт.

10. Забезпечення якості ПЗ.
11. Стандартизація розроблення ПЗ.
12. Міжнародні стандарти ISO.
13. Базовий стандарт ISO 12207.
14. Зміст основних процесів ЖЦ ПЗ.
15. Стандарт на процеси життєвого циклу систем ISO/IEC 15288.
16. Стандарти якості серії ISO 9000.
17. CASE–засоби та нотації моделювання програмних систем.
18. Візуальне моделювання мовою UML.
19. Базові терміни та нотація UML.
20. Методології розроблення ПЗ.
21. Вимоги щодо розроблення програм у RUP.
22. Методологія Microsoft Solution Framework (MSF).
23. Методологія eXtreme Programming (XP).
24. Маніфест гнучкого розроблення (Agile manifesto).
25. Принципи гнучкого розроблення ПЗ.
26. Agile-методи.
27. Патерни проектування програмних систем.
28. Забезпечення надійності розробки ПЗ.
29. Складність програмної системи.
30. Моделі якості процесів конструювання.
31. Методи тестування.
32. Види тестування програмного забезпечення.
33. Рівні тестування.
34. Історія патернів.
35. Класифікація патернів.
36. Приклади патернів.

7. Тренінг з дисципліни

Тренінг з дисципліни «Технологія розробки програмного забезпечення» виконується самостійно кожним студентом. Під час тренінгу прикладну область обирає студент самостійно. Метою виконання тренінгу є оволодіння навичками застосування методів технології розробки програмного забезпечення при розв’язуванні прикладних проблем. Тренінг оформлюється згідно з встановленими вимогами.

Тренінг формується на основі лабораторних робіт за наступною структурою:

1. Процент робіт, виконаних розробником (по кожному виду звітів).
2. Процент програмного коду, написаного розробником.
3. Опис особистого внеску в розробку системи.
4. Опис реалізованих розробником алгоритмів й опанованих в ході розробки проекту технологій/систем.

№ з/п	Вид роботи	Порядок проведення тренінгу
1	Вступна частина	ознайомлення студентів з темою тренінгового заняття і видача завдання
2	Практична частина	виконання завдань студентами згідно з індивідуальним завданням; оформлення короткого звіту
3	Підведення підсумків	Презентація та оцінювання результатів виконаних завдань

8. Засоби оцінювання та методи демонстрування результатів навчання

У процесі вивчення дисципліни «Технологія розробки програмного забезпечення» використовуються наступні засоби оцінювання та методи демонстрування результатів навчання:

- поточне тестування та опитування;
- оцінювання результатів лабораторних робіт;
- оцінювання виконання завдань самостійної роботи;
- оцінювання виконання завдань тренінгу;
- екзамен.

9. Критерії, форми поточного та підсумкового контролю

Підсумковий бал (за 100-бальною шкалою) з дисципліни «Технологія розробки програмного забезпечення» визначається як середньозважена величина, залежно від питомої ваги кожної складової залікового кредиту:

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

Оцінка за “Поточне оцінювання” визначається як середнє арифметичне з оцінок, отриманих під час занять (тестові завдання під час лекцій та оцінки лабораторних робіт).

Модуль “Тренінг” визначається, як середнє арифметичне з оцінок, отриманих під час тренінгу.

Модуль “Самостійна робота” визначається, як середнє арифметичне з оцінок, отриманих під час виконання завдання самостійної роботи, а саме оцінка за теоретичні знання, представлення результатів та проведених досліджень.

Шкала оцінювання:

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

10. Інструменти, обладнання та програмне забезпечення, використання яких передбачає навчальна дисципліна

№	Найменування	Номер теми
1.	Мультимедійне обладнання	1-22

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

Основні:

1. Електронний курс з дисципліни «Технологія розробки програмного забезпечення» для студентів галузі знань 12 «Інформаційні технології» спеціальності 122 «Комп'ютерні науки» на платформі Moodle ЗУНУ
2. Методичні вказівки до виконання лабораторних робіт з дисципліни ”Технологія розробки програмного забезпечення” для студентів освітньо-професійної програми «Штучний інтелект» спеціальності 122 «Комп’ютерні науки» першого (бакалаврського) рівня вищої освіти . Укладачі: Дорош В.І., Ліп’яніна-Гончаренко Х.В. Тернопіль: ЗУНУ, 2024. 44 с.
3. Опорний конспект лекцій з дисципліни ”Технологія розробки програмного забезпечення” для студентів освітньо-професійної програми «Штучний інтелект» спеціальності 122 «Комп’ютерні науки» першого (бакалаврського) рівня вищої освіти. Укладачі: Дорош В.І., Ліп’яніна-Гончаренко Х.В. Тернопіль: ЗУНУ, 2024. 90 с.
4. Volker Gruhn Rüdiger Striemer The Essence of Software Engineering / Volker Gruhn Rüdiger Striemer. – Saint Philip Street Press, 2020. – 248 p.
5. Chhavi Raj Dosaj The Self-Taught Software Tester A Step By Step Guide to Learn Software Testing Using Real-Life Project / Chhavi Raj Dosaj. – Amazon Digital Services LLC - KDP Print US, 2020. – 217 p.
6. Liping Liu Requirements Modeling and Coding: An Object-Oriented

Approach / L. Liping. – World Scientific Publishing Europe Ltd, 2020. – 451 p.
7. UML Tutorial [Електронний ресурс]. – Режим доступу: URL: https://www.tutorialspoint.com/uml/index.htm.
8. Java Tutorial [Електронний ресурс]. – Режим доступу: URL: https://www.tutorialspoint.com/java/index.htm.
9. C++ Tutorial [Електронний ресурс]. – Режим доступу: URL: https://www.tutorialspoint.com/cplusplus/index.htm.

Додаткові:

10. ДСТУ 2226-93 Автоматизовані системи. Терміни та визначення.
11. ДСТУ 3330-96 Інформаційні технології. Система стандартів з баз даних. Еталонна модель керування даними.
12. ДСТУ 3626-97 Базові програмно-технічні комплекси локального рівня для розосереджених автоматизованих систем керування технологічними процесами. Загальні вимоги.
13. ДСТУ ISO 11442-2:2004 Документація на технічну продукцію. Автоматизоване оброблення технічної інформації. Частина 2. Документація оригіналів.
14. ДСТУ ISO 11442-3:2004 Документація на технічну продукцію. Автоматизоване оброблення технічної інформації. Частина 3. Стадії процесу проектування продукції.
15. ДСТУ ISO 11442-4:2004 Документація на технічну продукцію. Автоматизоване оброблення технічної інформації. Частина 4. Системи керування та пошуку документів
16. РД 50-680-88 Керівний документ по стандартизації. Методичні вказівки. Автоматизовані системи. Основні положення.
17. РД 50-34.698-90 Автоматизовані системи. Вимоги до змісту документів
18. Jeong, Y. U., Yoo, S., Kim, Y. H., & Shim, W. H. (2020). De-identification of facial features in magnetic resonance images: software development using deep learning technology. Journal of medical Internet research, 22(12), e22739.
19. Petrov, P., Ivanov, S., Dimitrov, P., Dimitrov, G. P., & Bychkov, O. (2021). Projects Management in Technology Start-ups for Mobile Software Development. Int. J. Interact. Mob. Technol., 15(7), 194-201.
20. Maruping, L. M., & Matook, S. (2020). The evolution of software development orchestration: current state and an agenda for future research. European Journal of Information Systems, 29(5), 443-457.
21. Ліп’яніна–Гончаренко, Х. (2024). Узагальнений принцип синтезу інформаційної технології інтелектуального аналізу соціально–економічних даних ТГ. Measuring and Computing Devices in Technological Processes, (1), 359–367. https://doi.org/10.31891/2219–9365–2024–77–48
22. Intelligent Method for Classifying the Level of Anthropogenic Disasters. Lipianina-Honcharenko K, Wolff C, Sachenko A, Kit I, Zahorodnia D. Intelligent Method for Classifying the Level of Anthropogenic Disasters. Big Data and Cognitive Computing. 2023; 7(3):157. https://doi.org/10.3390/bdcc7030157 Scopus Q2