

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

ЗАТВЕРДЖУЮ:

В.о.декана факультету
комп'ютерних інформаційних
технологій
Ігор ЯКИМЕНКО

“ ” 2023 р.

ЗАТВЕРДЖУЮ:

В.о. проректора з
науково-педагогічної роботи
Віктор ОСТРОВЕРХОВ

“ ” 2023 р.

ЗАТВЕРДЖУЮ:

Директор навчально-наукового
інституту новітніх освітніх
технологій
Святослав ПИГЕЛЬ

“ ” 2023 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 – „Комп’ютерні науки”

Кафедра інформаційно-обчислювальних систем і управління

| Форма навчання | Курс | Семестр | Лекції (год.) | Лабораторні заняття (год.) | ІРС (год.) | Тренінг, КПЗ (год.) | Самост. робота студ. (год.) | Разом (год.) | Екз. (сем.) |
|----------------|------|---------|---------------|----------------------------|------------|---------------------|-----------------------------|--------------|-------------|
| Денна | 3 | 6 | 28 | 42 | 4 | 10 | 66 | 150 | 6 |
| Заочна | 3 | 6 | 8 | 4 | — | — | 138 | 150 | 6 |

Тернопіль – ЗУНУ
2023

31.08.2023

Робоча програма складена на основі освітньо-професійної програми «Комп'ютерні науки» підготовки бакалавра галузі знань 12 «Інформаційні технології» спеціальності 122 «Комп'ютерні науки», затвердженої Вченою радою ЗУНУ (протокол № 9 від 26 травня 2021 р.).

Робочу програму склав викладач кафедри інформаційно-обчислювальних систем і управління Віталій ДОРОШ

Робоча програма затверджена на засіданні кафедри інформаційно-обчислювальних систем і управління, протокол № 1 від 29 серпня 2023 р.

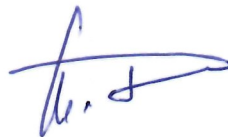
Завідувач кафедри



Мирослав КОМАР

Розглянуто та схвалено групою забезпечення спеціальності „Комп'ютерні науки”, протокол № 1 від 29 серпня 2023 р.

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



Мирослав КОМАР

Гарант освітньо-професійної
програми "Комп'ютерні науки",
к.т.н, доцент



Христина
ЛІП'ЯНИНА-ГОНЧАРЕНКО

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 – 5 | Галузь знань – 12 “Інформаційні технології” | Статус дисципліни: обов’язкова дисципліна циклу професійної підготовки Мова навчання: Українська |
| Кількість залікових модулів – 4 | Спеціальність – 122 «Комп’ютерні науки» | Рік підготовки: 3 Семестр: <i>Денна – 6</i> <i>Заочна – 6</i> |
| Кількість змістових модулів – 2 | Освітньо- професійна програма «Комп’ютерні науки» | Лекції: <i>Денна – 28 год.</i> <i>Заочна – 8 год.</i> Лабораторні заняття: <i>Денна – 42 год.</i> <i>Заочна – 4 год.</i> |
| Загальна кількість годин – 150 | Ступінь вищої освіти – бакалавр | Самостійна робота: <i>Денна – 66 год.,</i> <i>Заочна – 138 год.</i> Тренінг, КПЗ: <i>Денна – 10 год.</i> Індивідуальна робота: <i>Денна – 4 год.</i> |
| Тижневих годин – 10, з них аудиторних – 5 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:

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.

СК14. Здатність застосовувати методи та засоби забезпечення інформаційної безпеки, розробляти й експлуатувати спеціальне програмне забезпечення захисту інформаційних ресурсів об'єктів критичної інформаційної інфраструктури.

2.4. Передумови для вивчення дисципліни

Вивчення курсу „Технологія розробки програмного забезпечення” передбачає наявність систематичних та ґрунтовних знань із дисциплін: «Основи комп'ютерних наук», «Основи програмування», «Алгоритми і структури даних», «Сучасні парадигми програмування», «Бази і сховища даних», «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).

ПР15. Розуміти концепцію інформаційної безпеки, принципи безпечного проектування програмного забезпечення, забезпечувати безпеку комп'ютерних мереж в умовах неповноти та невизначеності вихідних даних.

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 проектування 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. Проектування ПЗ з допомогою структурних та поведінкових діаграм UML.

Діаграма послідовності (Sequence diagram). Діаграма пакетів (Package Diagram). Діаграма компонентів (Component Diagram). Діаграма розгортання

(Deployment Diagram).

Тема 9. Якість ПЗ.

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

Тема 10. Проектування UI.

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

Тема 11. 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 (Concrete Table Inheritance).

Тема 12. Чистий код.

Поняття чистого коду. Стандарти створення чистого коду: змістовні імена, стандарти написання функцій, коментарів, класів. Оброблення помилок.

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). Пристосованець (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. Супровід програмного забезпечення

Сутність та методи супроводу програмних систем. Завдання супроводу програмних систем. Якість супроводу програмних систем. Управління супроводом програмних систем.

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

Денна форма навчання

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

| | | | | | | |
|---|----|----|---|----|----|----------------------------|
| Тема 9. Якість ПЗ. | 2 | - | - | | 3 | Опитування під час заняття |
| Тема 10. Проектування UI. | 1 | 4 | 1 | | 3 | Опитування під час заняття |
| Тема 11. Об'єктні моделі та реляційні бази даних. | 1 | - | - | | 3 | Опитування під час заняття |
| Тема 12. Чистий код. | 1 | - | - | | 3 | Опитування під час заняття |
| Тема 13. Модульне тестування. | 2 | 4 | - | | 3 | Опитування під час заняття |
| <i>Змістовий модуль 2 – Керування конфігурацією, архітектура тестування та супровід ПЗ</i> | | | | | | |
| Тема 14. Системи контролю версій | 1 | 4 | - | 5 | 3 | Опитування під час заняття |
| Тема 15. Безперервна інтеграція та розгортання | 1 | - | - | | 3 | Опитування під час заняття |
| Тема 16. Принципи проектування | 1 | - | - | | 3 | Опитування під час заняття |
| Тема 17. Архітектурні стилі | 1 | - | 1 | | 3 | Опитування під час заняття |
| Тема 18. Багатошарова архітектура | 1 | - | - | | 3 | Опитування під час заняття |
| Тема 19. Породжуючі шаблони проектування | 1 | - | - | | 3 | Опитування під час заняття |
| Тема 20. Структурні шаблони проектування | 1 | - | - | | 3 | Опитування під час заняття |
| Тема 21. Поведінкові шаблони проектування | 1 | - | - | | 3 | Опитування під час заняття |
| Тема 22. Супровід програмного забезпечення | 1 | - | - | | 3 | Опитування під час заняття |
| Разом | 28 | 42 | 4 | 10 | 66 | |

Заочна форма навчання

| Тема | Кількість годин | | | |
|---|-----------------|---------------------|--------------|-------------------|
| | Лекції | Лабораторні заняття | Тренінг, КПЗ | Самостійна робота |
| <i>Змістовий модуль 1 - Загальні особливості розробки програмного забезпечення</i> | | | | |
| Тема 1. Базові поняття створення програмних продуктів | 4 | 2 | - | 6 |
| Тема 2. Класифікація та сутність процесів розроблення програмного забезпечення | | | | 6 |
| Тема 3. Моделі життєвого циклу ПЗ. | | | | 8 |
| Тема 4. Гнучкі методології. | | | | 8 |
| Тема 5. Інженерія вимог до ПЗ. | | | | 6 |
| Тема 6. Аналіз вимог до ПЗ. | | | | 6 |
| Тема 7. Проектування програмного забезпечення. | | | | 8 |
| Тема 8. Проектування ПЗ з допомогою структурних та поведінкових діаграм UML. | | | | 6 |
| Тема 9. Якість ПЗ. | | | | 6 |
| Тема 10. Проектування UI. | | | | 6 |
| Тема 11. Об'єктні моделі та реляційні бази даних. | | | | 6 |
| Тема 12. Чистий код. | | | | 6 |
| Тема 13. Модульне тестування. | | | | 6 |
| <i>Змістовий модуль 2 – Керування конфігурацією, архітектура тестування та супровід ПЗ</i> | | | | |
| Тема 14. Системи контролю версій | 4 | 2 | - | 6 |
| Тема 15. Безперервна інтеграція та розгортання | | | | 6 |
| Тема 16. Принципи проектування | | | | 6 |
| Тема 17. Архітектурні стилі | | | | 6 |
| Тема 18. Багатошарова архітектура | | | | 6 |
| Тема 19. Породжуючі шаблони проектування | | | | 6 |
| Тема 20. Структурні шаблони проектування | | | | 6 |
| Тема 21. Поведінкові шаблони проектування | | | | 6 |
| Тема 22. Супровід програмного забезпечення | | | | 6 |
| Разом | | | | 8 |

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. Комплексне практичне індивідуальне завдання

Індивідуальні завдання з дисципліни «Технологія розробки програмного забезпечення» виконується самостійно кожним студентом. КПЗ є науковим дослідження за варіантами, прикладну область обирає студент самостійно. Метою виконання КПЗ є оволодіння навичками застосування методів технології розробки програмного забезпечення при розв'язуванні прикладних проблем. КПЗ оформлюється згідно з встановленими вимогами.

КПЗ 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.

7. Самостійна робота

| № з/п | Тематика | Кількість годин | |
|---------------|--|-----------------|-----|
| | | ДФН | ЗФН |
| 1 | Базові поняття створення програмних продуктів | 3 | 6 |
| 2 | Класифікація та сутність процесів розроблення програмного забезпечення | 3 | 6 |
| 3 | Моделі життєвого циклу ПЗ. | 3 | 8 |
| 4 | Гнучкі методології. | 3 | 8 |
| 5 | Інженерія вимог до ПЗ. | 3 | 6 |
| 6 | Аналіз вимог до ПЗ. | 3 | 6 |
| 7 | Проектування програмного забезпечення. | 3 | 8 |
| 8 | Проектування ПЗ з допомогою структурних та поведінкових діаграм UML. | 3 | 6 |
| 9 | Якість ПЗ. | 3 | 6 |
| 10 | Проектування UI. | 3 | 6 |
| 11 | Об'єктні моделі та реляційні бази даних. | 3 | 6 |
| 12 | Чистий код. | 3 | 6 |
| 13 | Модульне тестування. | 3 | 6 |
| 14 | Системи контролю версій | 3 | 6 |
| 15 | Безперервна інтеграція та розгортання | 3 | 6 |
| 16 | Принципи проектування | 3 | 6 |
| 17 | Архітектурні стилі | 3 | 6 |
| 18 | Багатошарова архітектура | 3 | 6 |
| 19 | Породжуючі шаблони проектування | 3 | 6 |
| 20 | Структурні шаблони проектування | 3 | 6 |
| 21 | Поведінкові шаблони проектування | 3 | 6 |
| 22 | Супровід програмного забезпечення | 3 | 6 |
| Разом: | | 66 | 138 |

8. Тренінг з дисципліни

| № 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 |

9. 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;
- ректорська контрольна робота;
- оцінювання виконання завдань тренінгу;
- оцінювання результатів КПЗ;
- екзамен.

10. 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:

| Заліковий модуль 1 | Заліковий модуль 2 | Заліковий модуль 3 | Екзамен |
|---|---|---|--|
| 20 % | 20 % | 20 % | 40 % |
| 1. Виконання та захист лабораторних робіт (4 роботи по 10 балів) – 40 балів 2. Модульна контрольна робота – 60 балів | 1. Виконання та захист лабораторних робіт (4 роботи по 10 балів) – 40 балів 2. Ректорська контрольна робота – 60 балів | 1. Виконання завдань під час тренінгу – 20 балів 2. Написання та захист КПЗ – 80 балів | 1. Тестові завдання (25 тестів по 2 бали) – 50 балів 2. Завдання 1 – 25 балів 3. Завдання 2 – 25 балів |

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) |

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

| № | Найменування | Номер теми |
|----|--------------------------|------------|
| 1. | Мультимедійне обладнання | 1-22 |

РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

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