МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

Факультет МАТЕМАТИЧНИЙ

КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

**ЗАТВЕРДЖУЮ**

Декан математичного факультету

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ С. І. Гоменюк

«\_\_\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2022 р.

**АРХІТЕКТУРА ТА ПРОЄКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

підготовки бакалаврів

очної (денної) та заочної (дистанційної) форм здобуття освіти

спеціальності 121 «Інженерія програмного забезпечення»

освітньо-професійної програми «Програмна інженерія»

**Укладач:** Мильцев О.М., к.ф.-м.н., доцент кафедри програмної інженерії

|  |  |
| --- | --- |
| Обговорено та ухваленона засіданні кафедри програмної інженеріїПротокол № 1 від «01» вересня 2022 р.Завідувач кафедри \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А. О. Лісняк | Ухвалено науково-методичною радою математичного факультету Протокол № 1 від «01» вересня 2022 р.Голова науково-методичної ради факультету \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О. С. Пшенична |

|  |  |
| --- | --- |
| Погоджено з навчально-методичним відділом\_\_\_\_\_\_\_\_\_\_\_\_\_\_ О. В. Лещинська |  |

2022 рік

# Опис навчальної дисципліни

|  |  |  |
| --- | --- | --- |
| **1** | **2** | **3** |
| **Галузь знань, спеціальність,освітня програмарівень вищої освіти**  | **Нормативні показники для планування і розподілу дисципліни на змістові модулі**  | **Характеристика навчальної дисципліни** |
| очна (денна) форма здобуття освіти | заочна (дистанційна) форма здобуття освіти |
| **Галузь знань**12 Інформаційні технології | Кількість кредитів – 3 | **Обов’язкова** |
| **Цикл професійної підготовки** |
| **Спеціальність**121 Інженерія програмного забезпечення | Загальна кількість годин – 90 | **Семестр:** |
| 3-й |  |
| **Освітньо-професійна програма**Програмна інженерія | Змістових модулів – 4 | **Лекції** |
| 12 год. |  |
| **Лабораторні**  |
| 24 год. |  |
| **Рівень вищої освіти:** бакалаврський | Кількість поточних контрольних заходів – 22 | **Самостійна робота** |
| 34 год. |  |
| **Вид підсумкового семестрового контролю**: екзамен |

# Мета та завдання навчальної дисципліни

**Метою вивчення** навчальної дисципліни «Архітектура та проєктування програмного забезпечення» є отримання студентами теоретичних знань та набуття умінь з основних принципів проектування програмного забезпечення, шаблонів проектування, середовищ розробки і архітектур програмного забезпечення.

**Основними завданнями** вивчення дисципліни «Архітектура та проєктування програмного забезпечення» є оволодіння різними архітектурами програмного забезпечення, основними принципами та методами проєктування програмного забезпечення, програмними засобами проєктування програмного забезпечення, уніфікованою мовою проектування UML.

У результаті вивчення навчальної дисципліни студент повинен набути таких результатів навчання (знання, уміння тощо) та компетентностей:

| **Заплановані робочою програмою результати навчання та компетентності** | **Методи і контрольні заходи, що забезпечують досягнення результатів навчання та компетентностей** |
| --- | --- |
| **Програмні компетентності** |
| КЗ 1. Здатність до абстрактного мислення, аналізу та синтезу. | Методи навчання: лекція-візуалізація, майстер-класи, пояснення, дискусія, ілюстрація, робота з літературою, дослідницький метод, відповіді на запитання, лабораторні роботи, відеоконференції, відеозаписи лекцій.Контрольні заходи: захист лабораторної роботи, опитування, тестування. |
| КС 1. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення. | Методи навчання: лекція-візуалізація, майстер-класи, пояснення, дискусія, ілюстрація, робота з літературою, дослідницький метод, відповіді на запитання, лабораторні роботи, відеоконференції, відеозаписи лекцій.Контрольні заходи: захист лабораторної роботи, опитування, тестування. |
| КС 2. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування. | Методи навчання: лекція-візуалізація, майстер-класи, пояснення, дискусія, ілюстрація, робота з літературою, дослідницький метод, відповіді на запитання, лабораторні роботи, відеоконференції, відеозаписи лекцій.Контрольні заходи: захист лабораторної роботи, опитування, тестування. |
| КС 3. Здатність розробляти архітектури, модулі та компоненти програмних систем. | Методи навчання: лекція-візуалізація, майстер-класи, пояснення, дискусія, ілюстрація, робота з літературою, дослідницький метод, відповіді на запитання, лабораторні роботи, відеоконференції, відеозаписи лекцій.Контрольні заходи: захист лабораторної роботи, опитування, тестування. |
| КС 11. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення. | Методи навчання: лекція-візуалізація, майстер-класи, пояснення, дискусія, ілюстрація, робота з літературою, дослідницький метод, відповіді на запитання, лабораторні роботи, відеоконференції, відеозаписи лекцій.Контрольні заходи: захист лабораторної роботи, опитування, тестування. |
| **Програмні результати навчання** |
| ПР 1. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки. | Методи навчання: лекція-візуалізація, майстер-класи, пояснення, дискусія, ілюстрація, робота з літературою, дослідницький метод, відповіді на запитання, лабораторні роботи, відеоконференції, відеозаписи лекцій.Контрольні заходи: захист лабораторної роботи, опитування, тестування. |
| ПР 3. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення. | Методи навчання: лекція-візуалізація, майстер-класи, пояснення, дискусія, ілюстрація, робота з літературою, дослідницький метод, відповіді на запитання, лабораторні роботи, відеоконференції, відеозаписи лекцій.Контрольні заходи: захист лабораторної роботи, опитування, тестування. |
| ПР 6. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення. | Методи навчання: лекція-візуалізація, майстер-класи, пояснення, дискусія, ілюстрація, робота з літературою, дослідницький метод, відповіді на запитання, лабораторні роботи, відеоконференції, відеозаписи лекцій.Контрольні заходи: захист лабораторної роботи, опитування, тестування. |
| ПР 9. Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення. | Методи навчання: лекція-візуалізація, майстер-класи, пояснення, дискусія, ілюстрація, робота з літературою, дослідницький метод, відповіді на запитання, лабораторні роботи, відеоконференції, відеозаписи лекцій.Контрольні заходи: захист лабораторної роботи, опитування, тестування. |
| ПР 11. Вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання. | Методи навчання: лекція-візуалізація, майстер-класи, пояснення, дискусія, ілюстрація, робота з літературою, дослідницький метод, відповіді на запитання, лабораторні роботи, відеоконференції, відеозаписи лекцій.Контрольні заходи: захист лабораторної роботи, опитування, тестування. |
| ПР 12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення. | Методи навчання: лекція-візуалізація, майстер-класи, пояснення, дискусія, ілюстрація, робота з літературою, дослідницький метод, відповіді на запитання, лабораторні роботи, відеоконференції, відеозаписи лекцій.Контрольні заходи: захист лабораторної роботи, опитування, тестування. |
| ПР 14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення. | Методи навчання: лекція-візуалізація, майстер-класи, пояснення, дискусія, ілюстрація, робота з літературою, дослідницький метод, відповіді на запитання, лабораторні роботи, відеоконференції, відеозаписи лекцій.Контрольні заходи: захист лабораторної роботи, опитування, тестування. |
| ПР 17. Вміти застосовувати методи компонентної розробки програмного забезпечення. | Методи навчання: лекція-візуалізація, майстер-класи, пояснення, дискусія, ілюстрація, робота з літературою, дослідницький метод, відповіді на запитання, лабораторні роботи, відеоконференції, відеозаписи лекцій.Контрольні заходи: захист лабораторної роботи, опитування, тестування. |

**Міждисциплінарні зв’язки.**

Навчальна дисципліна «Архітектура та проєктування програмного забезпечення» застосовує досвід, отриманий здобувачами вищої освіти під час вивчення дисциплін «Основи програмної інженерії», «Бази даних», «Аналіз вимог до програмного забезпечення». Набуті при вивченні даної навчальної дисципліни знання та навички необхідні для подальшого вивчення дисциплін «Тестування, верифікація та валідація програмного забезпечення», «Моделювання та аналіз програмного забезпечення», «Стандартизація та якість програмного забезпечення», «Людино-машинний інтерфейс», «Виробнича практика».

# Програма навчальної дисципліни

**Змістовий модуль 1. Основи проєктування програмного забезпечення**

**Тема 1. Сучасні технології об’єктно-орієнтованого аналізу та проєктування програмного забезпечення**

П’ять ознак складної системи. Програмне забезпечення (ПЗ) як складна система. Роль декомпозиції в боротьбі зі складністю. Алгоритмічна декомпозиція. Об’єктно-орієнтована декомпозиція. Вибір між алгоритмічною та об’єктно-орієнтованою декомпозицією. Об’єктно-орієнтована парадигма. Інкапсуляція. Успадкування. Поліморфізм.

**Тема 2. Архітектура програмного забезпечення**

Поняття програмної архітектури та архітектурно-економічний цикл. Варіанти архітектури. Впливи на архітектуру зацікавлених осіб: фірми-розробника, архітектора, технічної бази. Зворотній вплив архітектури на фактори впливу на неї. Етапи розробки архітектури.

**Тема 3. Раціональний уніфікований процес розробки RUP**

П’ять представлень системи в UML. Шість основних моделей уніфікованого процесу розробки. Ітеративно-інкрементальна розробка. Маніфест гнучкої розробки програмного забезпечення. Візуальне моделювання.

**Тема 4. Проектування в UML**

Витоки Unified Modeling Language – UML. Три рівня моделей UML: концептуальні, логічні, фізичні. Діаграми UML: діаграми прецедентів, послідовностей, кооперацій, класів, діяльностей, станів, компонентів, розгортання. Використання UML на різних фазах проектування.

**Змістовий модуль 2.  Проєктування на концептуальному рівні в UML**

**Тема 5. Діаграми прецедентів**

Призначення діаграми. Елементи діаграми: актори, прецеденти, зв’язки. Типи зв’язків: асоціація, залежність включення, залежність розширення, узагальнення. Специфікація акторів. Специфікація прецедентів. Специфікація зв’язків. Пакети прецедентів. Документування потоку подій прецедентів.

**Змістовий модуль 3. Проєктування на логічному рівні в UML**

**Тема 6. Діаграми взаємодії: послідовності та кооперації**

Діаграми взаємодії. Призначення діаграми послідовності. Елементи діаграми послідовності: об’єкти і ролі, повідомлення. Типи повідомлень. Специфікація об’єктів. Специфікація повідомлень. Встановлення синхронізації та частоти повідомлень. Скрипти повідомлень. Співвіднесення повідомлення з операцією класу. Призначення діаграми кооперації. Елементи діаграми кооперації. Перемикання між діаграмою послідовності та діаграмою кооперації.

**Тема 7. Діаграми класів**

Призначення діаграми. Діаграма класів як головна діаграма UML. Елементи діаграми: класи, зв’язки. Клас на діаграмі класів. Типи класів. Іменування класів. Стереотипи класів. Видимість класів. Множинність класу. Стійкість класу. Паралелізм класу. Абстрактний клас. Вкладений клас. Пакети класів. Атрибути класів. Операції класів. Зв’язки між класами.

**Змістовий модуль 4. Проєктування на фізичному рівні в UML**

**Тема 8. Діаграми діяльності**

Призначення діаграми. Елементи діаграми: діяльність, вузол діяльності, потоки керування, розгалуження, поділ і з’єднання, паралельні потоки, «плаваючі доріжки», потік об’єктів. Види діаграм, що відображають діяльність. Інструменти на панелі діаграми діяльності. Приклади використання доріжок. Workflow. Реінжиніринг бізнес-процесів.

**Тема 9. Діаграми компонентів**

Призначення діаграми. Елементи діаграми. Типи компонентів. Компонентне програмування. Загальна характеристика. Адаптація доданку. Використання бібліотек компонентів. Розподілені компоненти. Вимоги до компонентів. Приклад компоненту ActiveX Data Objects (ADO). Порти. Артефакти.

**Тема 10. Діаграми розгортання**

Призначення діаграми. Елементи діаграми. Співвідношення діаграм класів, компонентів та розгортання. Вузли та артефакти. Організація вузлів.

# Структура навчальної дисципліни

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Змістовий модуль | Усього годин | Аудиторні (контактні) години | Самостійна робота, год | Система накопичення балів |
| Усього годин | Лекційні заняття, год. | Лабораторні заняття, год. | о/д ф. | з/дистф. | Теор.завд.,к-ть балів | Лаб.завд.,к-ть балів | Усього балів |
| о/д ф. | з/дист ф. | о/д ф. | з/дист ф. | о/д ф. | з/дистф. |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** |
| 1 | 22 | 10 |  | 4 |  | 6 |  | 12 |  | 3 | 12 | 15 |
| 2 | 22 | 8 |  | 2 |  | 6 |  | 14 |  | 7 | 8 | 15 |
| 3 | 24 | 10 |  | 4 |  | 6 |  | 14 |  | 2 | 8 | 10 |
| 4 | 22 | 8 |  | 2 |  | 6 |  | 14 |  | 8 | 12 | 20 |
| Усього за змістові модулі | **90** | **36** | **0** | **12** |  | **24** |  | **54** |  | **20** | **40** | **60** |
| Підсумковий семестровий контроль**екзамен** |  |  |  |  |  |  |  |  |  | **20** | **20** | **100** |
| Загалом | **90** | **36** | **0** | **12** | **0** | **24** | **0** | **54** | **0** | **40** | **60** | **100** |

# Теми лекційних занять

|  |  |  |
| --- | --- | --- |
| **№ змістового модуля** | **Назва теми** | **Кількість годин** |
| **о/д ф.** | **з/дист ф.** |
| 1 | Сучасні технології об’єктно-орієнтованого аналізу та проектування програмного забезпечення.Архітектура програмного забезпечення. | 2 |  |
| Раціональний уніфікований процес розробки RUP.Проектування в UML. | 2 |  |
| 2 | Діаграми прецедентів | 2 |  |
| 3 | Діаграми взаємодії: послідовності та кооперації. | 2 |  |
| Діаграми класів. | 2 |  |
| 4 | Діаграми діяльності.Діаграми компонентів.Діаграми розгортання. | 2 |  |
|  | **Разом** | **12** |  |

# Теми лабораторних занять

|  |  |  |
| --- | --- | --- |
| **№ змістового модуля** | **Назва теми** | **Кількість годин** |
| **о/д ф.** | **з/дист ф.** |
| 1 | Приклади складних систем та застосування основних принципів об'єктно-орієнтованого підходу. | 2 |  |
| Формулювання призначення і цілей створення інформаційної системи. | 2 |  |
| Визначення переліку акторів та прецедентів. | 2 |  |
| 2 | Опис прецедентів. | 4 |  |
| Діаграми прецедентів. | 2 |  |
| 3 | Діаграми взаємодії: послідовності та кооперації. | 4 |  |
| Діаграми класів. | 2 |  |
| 4 | Діаграми діяльності. | 2 |  |
| Діаграми компонентів. | 2 |  |
| Діаграми розгортання. | 2 |  |
|  | **Разом** | **24** |  |

# Види і зміст поточних контрольних заходів

| **№ змістового модуля** | **Вид поточного контрольного заходу** | **Зміст поточного контрольного заходу** | **Критерії оцінювання** | **Усього балів** |
| --- | --- | --- | --- | --- |
| 1 | Звіт з виконання лабораторної роботи 1 | Перевірка виконання завдань лабораторної роботи, оволодіння теоретичним матеріалом | Критерії до захисту лабораторних робіт | **4** |
| Усне опитування і обговорення контрольних питань лабораторної роботи 1 | Згідно змісту лекційного матеріалу, та матеріалу, що виноситься на самостійну роботу | Критерії до теоретичного та практичного поточного контролю | **1** |
| Звіт з виконання лабораторної роботи 2 | Перевірка виконання завдань лабораторної роботи, оволодіння теоретичним матеріалом | Критерії до захисту лабораторних робіт | **4** |
| Усне опитування і обговорення контрольних питань лабораторної роботи 2 | Згідно змісту лекційного матеріалу, та матеріалу, що виноситься на самостійну роботу | Критерії до теоретичного та практичного поточного контролю | **1** |
| Звіт з виконання лабораторної роботи 3 | Перевірка виконання завдань лабораторної роботи, оволодіння теоретичним матеріалом | Критерії до захисту лабораторних робіт | **4** |
| Усне опитування і обговорення контрольних питань лабораторної роботи 3 | Згідно змісту лекційного матеріалу, та матеріалу, що виноситься на самостійну роботу | Критерії до теоретичного та практичного поточного контролю | **1** |
| 2 | Звіт з виконання лабораторної роботи 4 | Перевірка виконання завдань лабораторної роботи, оволодіння теоретичним матеріалом | Критерії до захисту лабораторних робіт | **4** |
| Усне опитування і обговорення контрольних питань лабораторної роботи 4 | Згідно змісту лекційного матеріалу, та матеріалу, що виноситься на самостійну роботу | Критерії до теоретичного та практичного поточного контролю | **1** |
| Звіт з виконання лабораторної роботи 5 | Перевірка виконання завдань лабораторної роботи, оволодіння теоретичним матеріалом | Критерії до захисту лабораторних робіт | **4** |
| Усне опитування і обговорення контрольних питань лабораторної роботи 5 | Згідно змісту лекційного матеріалу, та матеріалу, що виноситься на самостійну роботу | Критерії до теоретичного та практичного поточного контролю | **1** |
| Тестування за пройденим матеріалом змістовних модулів 1–2 |  |  | **5** |
| 3 | Звіт з виконання лабораторної роботи 6 | Перевірка виконання завдань лабораторної роботи, оволодіння теоретичним матеріалом | Критерії до захисту лабораторних робіт | **4** |
| Усне опитування і обговорення контрольних питань лабораторної роботи 6 | Згідно змісту лекційного матеріалу, та матеріалу, що виноситься на самостійну роботу | Критерії до теоретичного та практичного поточного контролю | **1** |
| Звіт з виконання лабораторної роботи 7 | Перевірка виконання завдань лабораторної роботи, оволодіння теоретичним матеріалом | Критерії до захисту лабораторних робіт | **4** |
| Усне опитування і обговорення контрольних питань лабораторної роботи 7 | Згідно змісту лекційного матеріалу, та матеріалу, що виноситься на самостійну роботу | Критерії до теоретичного та практичного поточного контролю | **1** |
| 4 | Звіт з виконання лабораторної роботи 8 | Перевірка виконання завдань лабораторної роботи, оволодіння теоретичним матеріалом | Критерії до захисту лабораторних робіт | **4** |
| Усне опитування і обговорення контрольних питань лабораторної роботи 8 | Згідно змісту лекційного матеріалу, та матеріалу, що виноситься на самостійну роботу | Критерії до теоретичного та практичного поточного контролю | **1** |
| Звіт з виконання лабораторної роботи 9 | Перевірка виконання завдань лабораторної роботи, оволодіння теоретичним матеріалом | Критерії до захисту лабораторних робіт | **4** |
| Усне опитування і обговорення контрольних питань лабораторної роботи 9 | Згідно змісту лекційного матеріалу, та матеріалу, що виноситься на самостійну роботу | Критерії до теоретичного та практичного поточного контролю | **1** |
| Звіт з виконання лабораторної роботи 10 | Перевірка виконання завдань лабораторної роботи, оволодіння теоретичним матеріалом | Критерії до захисту лабораторних робіт | **4** |
| Усне опитування і обговорення контрольних питань лабораторної роботи 10 | Згідно змісту лекційного матеріалу, та матеріалу, що виноситься на самостійну роботу | Критерії до теоретичного та практичного поточного контролю | **1** |
| Тестування за пройденим матеріалом змістовних модулів 3–4 |  |  | **5** |
| **Усього** | **60** |

Теоретичні відомості та варіанти до лабораторних робіт розміщуються на платформі Moodle за посиланням: <https://moodle.znu.edu.ua/course/view.php?id=16>

До кожної лабораторної роботи потрібно скласти звіт про її виконання, який пояснює всі етапи виконання роботи. Звіт складається в електронному вигляді за вимогами, які висуваються до оформлення курсових і кваліфікаційних робіт для здобувачів ступеня вищої освіти бакалавра та магістра математичного факультету. Якість оформлення звіту враховується при оцінюванні роботи. Файли вихідного коду або проекту, що реалізує завдання роботи, додаються до звіту окремим файлом.

Захист лабораторної роботи є обов’язковим і потребує пояснення всіх етапів виконання завдання. .

Оцінювання звіту про виконання лабораторної роботи і усного теоретичного опитування при захисті цього завдання здійснюється за формулою

*s = m \* v / 100*, (1)

де *s* – підсумковий бал за вид контролю, *m* – максимальний бал за вид контролю, *v* - відсоток виконання.

Критерії визначення *v* (%):

• 90-100%: контрольний захід здійснено без помилок; це відповідає виявленню студентом всебічного системного і глибокого знання програмного матеріалу; засвоєнню ним основної і додаткової літератури; чіткому володінню понятійним апаратом, методами, методиками та інструментами, передбаченими програмою дисципліни; вмінню використовувати їх для вирішення як типових, так і нетипових практичних ситуацій; виявленню творчих здібностей в розумінні, викладі та використанні навчально-програмного матеріалу;

• 60-89%: контрольний захід здійснено без суттєвих помилок; відповідає виявленню знань основного програмного матеріалу; засвоєнню інформації в межах лекційного курсу; володінню необхідними методами, методиками та інструментами, передбаченими програмою; вмінню використовувати їх для вирішення типових ситуацій, припускаючи окремих незначних помилок;

• 0-59%: більше 30% контрольний захід здійснено невірно; відповідає виявленню значних прогалин у знаннях основного програмного матеріалу; не досить упевненому володінню окремими поняттями, методиками та інструментами, про що свідчать принципові помилки під час їх використання.

# Підсумковий семестровий контроль

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Форма** | **Види підсумкових контрольних заходів** | **Зміст підсумкового контрольного заходу** | **Критерії оцінювання** | **Усього балів** |
| **Залік** | Екзаменаційний тест | Підсумковий тест на платформі Moodle Тест складається з 20 завдань: 16 теоретичних (тестові питання з вибором правильної відповіді з декількох можливих), 4 практичних (розв’язання задач з вибором правильної відповіді з декількох можливих) | Кожне теоретичне завдання оцінюється в 2 бал, практичні завдання: 2 бали | **40** |

# Рекомендована література

**Основна:**

1. Лісняк А. О., Мильцев О. М., Мухін В. В., Чопорова О. В. Архітектура та проектування програмного забезпечення: методичні рекомендації до лабораторних занять для здобувачів ступеня вищої освіти бакалавра спеціальності 121 «Інженерія програмного забезпечення» освітньо-професійної програми «Програмна інженерія». Запоріжжя : ЗНУ, 2022. 53 с.
2. Роберт Мартін Чиста архітектура. Мистецтво розробки програмного забезпечення, 2019. 368с.
3. Edwin Mach Object Oriented Analysis & Design Cookbook: Introduction to Practical System Modeling, 2019. 218p.
4. Suriya Sundaramoorthy UML Diagramming: A Case Study Approach 1st Edition, 2022. 416p.
5. Lukman Adam Obomeghie, Bayo Adedeji, Sylvester Aketaumhe Unified Modeling Language Practical Guide, 2022. 59p.