

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ ЕКОНОМІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ СЕМЕНА КУЗНЕЦЯ**

ЗАТВЕРДЖЕНО

на засіданні кафедри
інформаційних систем.
Протокол № 1 від 22.08.2023 р.

ПОГОДЖЕНО

Проректор з навчально-методичної роботи

Каріна НЕМАШКАЛО



ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

робоча програма навчальної дисципліни (РПНД)

Галузь знань	12 "Інформаційні технології"
Спеціальність	121 "Інженерія програмного забезпечення"
Освітній рівень	перший (бакалаврський)
Освітня програма	"Інженерія програмного забезпечення"

Статус дисципліни	обов'язкова
Мова викладання, навчання та оцінювання	українська

Розробник:
к.е.н., професор

підписано КЕП

Ірина ЗОЛОТАРЬОВА

Завідувач кафедри
інформаційних систем

Дмитро БОНДАРЕНКО

Гарант програми

Олег ФРОЛОВ

Харків
2024

ВСТУП

Дисципліна присвячена вивченню методів і засобів управління програмними проектами (планування робіт і регулювання ресурсів), експертного оцінювання проміжних результатів розроблення під час процесів життєвого циклу (ЖЦ), оцінювання ризику побудови програмної системи і досягнутої для неї якості. Ця дисципліна використовує стандарти, що регламентують процеси ЖЦ, інженерію вимог, тестування і забезпечення якості шляхом перевірки показників на процесах ЖЦ і кінцевого продукту для їхнього оцінювання. Розглядаються питання теоретичної та практичної побудови різних програмних систем для виконання задач з розробки програмного забезпечення. Навчальна дисципліна буде корисна майбутнім керівникам і виконавцям проектів по розробці інформаційних систем, насамперед під час організації процесів проектування програмних продуктів сучасними засобами.

Навчальна дисципліна "Інженерія програмного забезпечення" вивчається здобувачами спеціальності «Інженерія програмного забезпечення» усіх форм навчання на третьому курсі протягом шостого семестру.

Метою викладання навчальної дисципліни "Інженерія програмного забезпечення" є вивчення здобувачами вищої освіти методологій для розробки програмного забезпечення інформаційних систем; застосування мови UML (Universal Modeling Language) для моделювання і проектування інформаційних систем; застосування програмного інструментарію розширення та поглиблення теоретичних знань і прикладних вмінь і навичок щодо основних понять та визначень з галузі забезпечення якості програмного забезпечення, придбання навичок застосування сучасних інформаційних технологій для аналізу інформаційних систем, створення звітної документації.

Формування системного розуміння:

Сучасних концепцій, принципів, методологій та технологій розробки програмного забезпечення (ПЗ).

Життєвого циклу (ЖЦ) програмних систем інформаційних систем (ІС).

Кращих практик та стандартів програмної інженерії (ПІ), що використовуються в ІТ-індустрії.

Оволодіння навичками:

Проектування, розробки, тестування та впровадження ПЗ.

Використання сучасних інструментальних засобів ПІ.

Застосування принципів ООП при проектуванні та розробці ПЗ.

Формулювання та документування вимог до ПЗ.

Забезпечення якості ПЗ згідно з заданими стандартами.

Вивчення:

Етапів, моделей та артефактів ЖЦ ПЗ.

Класів, об'єктів, методів, спадкування та поліморфізму в ООП.

Стандартів програмної інженерії IEEE, ISO/IEC, CMMI.

Методів та прийомів тестування ПЗ.

Забезпечення якості ПЗ: стандарти, методи, метрики.

CASE-засобів, IDE, систем контролю версій.

Завданнями навчальної дисципліни є: формування у здобувачів систематизованого уявлення щодо понять, принципів, методології та технологій створення програмних продуктів як сукупності процесів розроблення програмних систем на засадах життєвого циклу (ЖЦ) програмного забезпечення інформаційних систем; вивченні призначення засобів об'єктного підходу до проектування програмного забезпечення (ПЗ); стандартів програмної інженерії; методів створення вимог під час розроблення програмних продуктів; методів та прийомів тестування програмного забезпечення сучасними засобами тестування програмних продуктів; стандартів якості під час створення програмного забезпечення інформаційних систем; інструментальних засобів програмної інженерії та їх практичне застосування під час проектування програмного забезпечення.

Предметом навчальної дисципліни є технології розроблення, тестування та експлуатації програмних модулів інформаційних систем на основі сучасних методології та засобів об'єктно-орієнтованого підходу для проектування програмних продуктів, вивчення методологій розроблення програмного забезпечення інформаційних систем, застосування мови Universal Modeling Language (UML) для моделювання і проектування ПЗ, застосування інструментарію – IBM Rational Rose, Scrum, Adile, Canban вивчення основних положень з проектування ПЗ на базі сучасних технічних, програмних, інструментальних та комунікаційних засобів.

Об'єктом навчальної дисципліни є процеси, що відображають різні аспекти створення та супроводу програмних продуктів.

Результати навчання та компетентності, які формує навчальна дисципліна визначено в табл. 1.

Таблиця 1

Результати навчання та компетентності, які формує навчальна дисципліна

Результати навчання	Компетентності, якими повинен оволодіти здобувач вищої освіти
PH01	ЗК3, ЗК5, ЗК6, СК8, СК10
PH02	ЗК2, ЗК8, ЗК9, ЗК10, ЗК11, ЗК12, СК5, СК9
PH03	ЗК7, СК11, СК12
PH04	СК4, СК5, СК12
PH06	ЗК1, ЗК2, СК10, СК11
PH07	ЗК1, ЗК2, СК6, СК7, СК8, СК11
PH12	СК1, СК2, СК3
PH13	СК8
PH14	ЗК2, ЗК3, СК2, СК3, СК13
PH16	СК15

де, PH01. Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.

PH02. Знати кодекс професійної етики, розуміти соціальну значимість та культурні аспекти інженерії програмного забезпечення і дотримуватись їх в професійній діяльності.

PH03. Знати основні процеси, фази та ітерації життєвого циклу програмного

забезпечення.

PH04. Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення.

PH06. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.

PH07. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

PH12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення;

PH13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.

PH14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.

PH16. Мати навички командної розробки, погодження, оформлення і випуску всіх видів програмної документації.

ЗК01. Здатність до абстрактного мислення, аналізу та синтезу.

ЗК02. Здатність застосовувати знання у практичних ситуаціях.

ЗК03. Здатність спілкуватися державною мовою як усно, так і письмово.

ЗК05. Здатність вчитися і оволодівати сучасними знаннями.

ЗК06. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.

ЗК07. Здатність працювати в команді.

ЗК08. Здатність діяти на основі етичних міркувань.

ЗК09. Прагнення до збереження навколишнього середовища.

ЗК10. Здатність діяти соціально відповідально та свідомо.

ЗК11. Здатність реалізувати свої права і обов'язки як члена суспільства, усвідомлювати цінності громадянського (вільного демократичного) суспільства та необхідність його сталого розвитку, верховенства права, прав і свобод людини і громадянина в Україні.

ЗК12. Здатність зберігати та примножувати моральні, культурні, наукові цінності і досягнення суспільства на основі розуміння історії та закономірностей розвитку предметної області, її місця у загальній системі знань про природу і суспільство та у розвитку суспільства, техніки і технологій, використовувати різні види та форми рухової активності для активного відпочинку та ведення здорового способу життя.

СК01. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.

СК02. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

СК03. Здатність розробляти архітектури, модулі та компоненти програмних систем.

СК04. Здатність формулювати та забезпечувати вимоги щодо якості програмного забезпечення у відповідності з вимогами замовника, технічним завданням та стандартами.

СК05. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу.

СК06. Здатність аналізувати, вибирати і застосовувати методи і засоби для забезпечення інформаційної безпеки (в тому числі кібербезпеки).

СК07. Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних.

СК08. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.

СК09. Здатність оцінювати і враховувати економічні, соціальні, технологічні та екологічні чинники, що впливають на сферу професійної діяльності.

СК10. Здатність накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя.

СК11. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.

СК12. Здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення.

СК13. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Зміст навчальної дисципліни

Тема 1. Моделі життєвого циклу програмного забезпечення. Документування автоматизованої програмної системи (АПС). Vision & Scope, SRS, User Stories.

Огляд основних моделей ЖЦ ПЗ (водоспадна, ітеративна, спіральна, інкрементна). Переваги та недоліки кожної моделі. Вибір оптимальної моделі ЖЦ ПЗ для конкретного проекту. Професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення.

Документування АПС. Види документації АПС (технічне завдання, специфікація вимог, експлуатаційна документація). Правила оформлення документації АПС. Використання CASE-засобів для документування АПС.

Планування проекту розробки ПЗ. Визначення цілей, завдань та етапів проекту. Оцінка ресурсів, необхідних для реалізації проекту, врахування економічних, соціальних, технологічних та екологічних чинників.

Управління ризиками проекту. Ідентифікація та аналіз ризиків проекту. Розробка плану реагування на ризики. Моніторинг та контроль ризиків проекту.

Тема 2. Software Requirements як область знань в програмній інженерії. Управління вимогами і зв'язок із задачами SWEBOOK.

Поняття та класифікація вимог до ПЗ: Функціональні та нефункціональні вимоги. Якісні та кількісні характеристики вимог.

Збір та аналіз вимог. Методи збору вимог (інтерв'ю, опитування, спостереження, аналіз документів). Аналіз повноти, правильності, узгодженості та пріоритетності вимог. Використання CASE-засобів для управління вимогами.

Документування вимог. Види документів, що описують вимоги (специфікація вимог, технічне завдання). Правила оформлення документів з вимогами. Використання шаблонів та стандартів для документування вимог.

Управління вимогами. Планування та контроль процесу збору, аналізу, документування та верифікації вимог. Зміна та верифікація вимог. Управління

конфліктами та ризиками, пов'язаними з вимогами. Аналіз та вибір методів і засобів забезпечення інформаційної безпеки.

Зв'язок з іншими областями знань SWEBOOK. Вплив вимог на інші етапи життєвого циклу ПЗ. Забезпечення відповідності ПЗ вимогам.

Тема 3. Особливості визначення та аналізу бізнес-вимог. Визначення вимог як етап розробки програмного забезпечення. Проблеми управління процесом розроблення вимог до програмного забезпечення.

Визначення бізнес-вимог. Процес визначення, аналізу та формулювання вимог до програмного продукту, що виходить з потреб бізнесу та його цілей. Застосування фундаментальних і міждисциплінарних знань для розв'язання завдань.

Аналіз бізнес-вимог. Процес аналізу бізнес-вимог, включаючи визначення структури вимог, їх пріоритетів та взаємозв'язків, а також розробку вимог до функціональності.

Етап розробки програмного забезпечення. Важливість визначення вимог на ранніх етапах розробки ПЗ, методика та підходи до цього процесу.

Проблеми управління процесом розроблення вимог. Проблеми, які можуть виникнути під час процесу розроблення вимог та способи їх вирішення.

Ключові аспекти управління вимогами: управління змінами вимог, управління конфліктами вимог, управління ризиками та інші.

Тема 4. Об'єктно-орієнтований підхід до проєктування програмного забезпечення. Мова UML.

Основні поняття об'єктно-орієнтованого підходу: базові концепції об'єктно-орієнтованого підходу, класи, об'єкти, спадкування, поліморфізм та інші.

Процес моделювання за допомогою UML. Основні елементи UML (Unified Modeling Language): діаграми класів, діаграми взаємодії, діаграми послідовності, діаграми діяльності та інші, а також їх застосування в процесі моделювання програмного забезпечення.

Розробка програмного забезпечення за допомогою UML. Методика та підходи до розробки програмного забезпечення з використанням UML, такі як Agile, Rational Unified Process (RUP), Scrum тощо.

Управління процесом розроблення програмного забезпечення з використанням UML. Проблеми та методи управління процесом розроблення програмного забезпечення з використанням UML

Тема 5. Процес аналізу вимог. Діаграма варіантів використання UML.

Процес аналізу вимог. Процес збору, аналізу та документування вимог до програмного забезпечення. Визначення функціональних та нефункціональних вимог, узгодження вимог зі стейкхолдерами, а також формулювання узагальнених вимог та вимог до проєкту.

Методи аналізу вимог. Прототипування, дослідницький метод, діаграми варіантів використання UML (Use Case Diagrams) тощо.

Діаграма варіантів використання UML (Use Case Diagrams). Концепція діаграм варіантів використання UML, їх структура та елементи, такі як актори, варіанти використання, включення, розширення тощо.

Приклади використання діаграм варіантів використання UML для моделювання та аналізу вимог до програмного забезпечення.

Тема 6. Методи об'єктного аналізу і моделювання. Розширений аналіз вимог. Діаграма діяльності UML.

Методи об'єктного аналізу і моделювання. Методи аналізу об'єктів, методи аналізу сценаріїв та інші.

Розширений аналіз вимог. Аналіз діаграми варіантів використання, аналіз умовностей, аналіз обмежень тощо.

Діаграма діяльності UML (Activity Diagrams), її структура, елементи, призначення та методи аналізу та моделювання діяльності.

Приклади використання діаграми діяльності UML для моделювання різних процесів та діяльностей в програмному забезпеченні.

Тема 7. Управління змінами вимог до ПЗ. Основні задачі управління вимогами. Трасування вимог.

Управління змінами вимог до ПЗ: визначення, документування та відстеження змін вимог, а також їх вплив на процес розробки.

Основні задачі управління вимогами: визначення, аналіз, документування, збереження та відстеження вимог протягом життєвого циклу розробки програмного забезпечення. Нефункціональні вимоги, враховуючі питання збереження навколишнього середовища, сталого розвитку, соціальної відповідальності, цінностей демократичного суспільства, верховенства права, прав людини та громадянина, відповідно до законів України, розуміння історичного контексту та закономірностей розвитку суспільства, техніки та технологій.

Трасування вимог: процес відстеження вимог від початкових документів та аналізу їх впливу на різні етапи розробки програмного забезпечення.

Процеси управління змінами вимог: процеси та методики управління змінами вимог, такі як Agile, Scrum, Waterfall, а також інструменти та практики їх впровадження.

Тема 8. Етап проєктування ПЗ. Діаграма станів UML.

Опис етапу проєктування ПЗ, включаючи аналіз вимог, архітектурне проєктування, детальне проєктування, моделювання та тестування.

Діаграма станів UML (Statechart Diagrams): її структура, елементи, призначення та методи аналізу та моделювання станів системи.

Процеси та методи проєктування ПЗ з використанням діаграми станів UML. Приклади використання діаграми станів UML для моделювання різних станів та переходів системи в програмному забезпеченні.

Тема 9. Проєктування архітектури системи. Діаграми класів та компонентів UML.

Опис процесу проєктування архітектури системи. Методи та процеси проєктування архітектури системи, такі як визначення основних компонентів, їх взаємозв'язків та інші.

Діаграма класів UML (Class Diagram): її структура, елементи, призначення та методи аналізу та моделювання класів.

Процеси та методи проектування архітектури системи з використанням діаграми класів UML. Приклади використання діаграми класів UML для моделювання різних класів та їх взаємозв'язків в програмному забезпеченні.

Діаграма компонентів UML (Component Diagram): її структура, елементи, призначення

Визначення процесу інтеграції системи, застосування стандартів і процедур управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення.

Тема 10. Розробники програмного забезпечення. Робота в команді. Основні функції розробника програмного забезпечення: аналіз вимог, проектування, реалізація, тестування та підтримка ПЗ.

Основні навички та знання, які повинні мати розробники ПЗ, такі як програмування, алгоритмізація, архітектура ПЗ, тестування, управління версіями, комунікація та інші.

Організація та робота в команді. Методи та підходи до організації роботи розробників в команді, такі як Agile, Scrum, Waterfall тощо, а також проблеми та підходи до їх вирішення.

Комунікація та спілкування в команді. Методи та підходи до спілкування та комунікації в команді, такі як зустрічі, спільне планування, обмін інформацією тощо. Підходи щодо накопичення, обробки та систематизації професійних знання щодо створення і супроводження програмного забезпечення, навчання протягом життя.

Завдання та відповідальності розробників ПЗ в команді.

Перелік лабораторних занять за навчальною дисципліною наведено в табл. 2.

Таблиця 2

Перелік лабораторних занять

Назва теми	Зміст
Тема 1-3.	Бізнес-аналіз (аналіз предметної області). 1) Опис поточної ситуації (AS IS) в форматі UML (діаграма варіантів використання) . 2) Опис бізнес-проблем замовника / клієнтів. 3) Опис зацікавлених осіб / кінцевих користувачів. 4) Словник предметної області (глосарій). 5) Аналіз бізнес-вимог у вигляді карти цілей та у вигляді карти впливів: Impact Mapping.
Тема 4.	Аналіз існуючих рішень (аналогів, конкурентів). Порівняльний аналіз за критеріями.
Тема 5-7.	Створення концепції рішення. Опис функцій рішення. Деталізація функціональних і нефункціональних вимог. Обґрунтування інструментарію для розробки та супроводження програмного забезпечення.
Тема 8-10.	Проектування рішення. Проектування архітектури системи. Прототипування інтерфейсу користувача. Рисунки екранів інтерфейсу / сторінок.

Перелік самостійної роботи за навчальною дисципліною наведено в табл. 3.

Таблиця 3

Перелік самостійної роботи

Назва теми	Зміст
Тема 1 – 10	Вивчення лекційного матеріалу
Тема 1 – 10	Підготовка до лабораторних занять
Тема 1 – 10	Підготовка до екзамену

Кількість годин лекційних та лабораторних занять, а також годин самостійної роботи наведено в робочому плані (технологічній карті) з навчальної дисципліни.

МЕТОДИ НАВЧАННЯ

У процесі викладання навчальної дисципліни для набуття визначених результатів навчання, активізації освітнього процесу передбачено застосування таких методів навчання, як:

Словесні (лекція (Тема 1, 3, 5, 7, 9, 10), проблемна лекція (Тема 8), лекція-візуалізація (Тема 2, 4, 6)).

Наочні (демонстрація (Тема 1-10)).

Лабораторна робота (Тема 1 – 4), проектний метод (Тема 1-4).

ФОРМИ ТА МЕТОДИ ОЦІНЮВАННЯ

Університет використовує 100 бальну накопичувальну систему оцінювання результатів навчання здобувачів вищої освіти.

Поточний контроль здійснюється під час проведення лекційних та лабораторних занять і має на меті перевірку рівня підготовленості здобувача вищої освіти до виконання конкретної роботи і оцінюється сумою набраних балів:

– для дисциплін з формою семестрового контролю екзамен (іспит): максимальна сума – 60 балів; мінімальна сума, що дозволяє здобувачу вищої освіти скласти екзамен (іспит) – 35 балів.

Підсумковий контроль включає семестровий контроль та атестацію здобувача вищої освіти.

Семестровий контроль проводиться у формі семестрового екзамену (іспиту). Складання семестрового екзамену (іспиту) здійснюється під час екзаменаційної сесії.

Максимальна сума балів, яку може отримати здобувач вищої освіти під час екзамену (іспиту) – 40 балів. Мінімальна сума, за якою екзамен (іспит) вважається складеним – 25 балів.

Підсумкова оцінка за навчальною дисципліною визначається сумуванням балів за поточний та підсумковий контроль.

Під час викладання навчальної дисципліни використовуються наступні контрольні заходи:

Поточний контроль: захист лабораторних робіт (40 балів), письмова контрольна робота (тестування) (20 балів).

Семестровий контроль: Екзамен (40 балів).

Більш детальну інформацію щодо системи оцінювання наведено в робочому плані (технологічній карті) з навчальної дисципліни.

Приклад екзаменаційного білета та критерії оцінювання для навчальної дисципліни.

Приклад екзаменаційного білета

Харківський національний економічний університет імені Семена Кузнеця
Перший (бакалаврський) рівень вищої освіти
Спеціальність «Інженерія програмного забезпечення»
Освітньо-професійна програма «Інженерія програмного забезпечення»
Семестр 6
Навчальна дисципліна «Інженерія програмного забезпечення»

ЕКЗАМЕНАЦІЙНИЙ БІЛЕТ № 1

Завдання 1

Розроблення автоматизованого модуля «Облік руху кадрів на підприємстві»

Розроблюваний автоматизований модуль пропонується для впровадження на приватних та державних підприємствах з метою підвищення продуктивності праці спеціалістів відділу кадрів. Досягнення поставленої мети здійснюється впровадженням автоматизованого рішення наступних задач:

- прийом працівників;
- переміщення працівників;
- звільнення працівників;
- аналіз кадрового стану підприємства.

У розроблюваному модулі необхідно передбачити ідентифікацію користувачів системи. Кінцевий користувач повинен мати можливість в повному обсязі виконувати операції з даними по робітникам, посадам, підрозділам, статтям звільнень, здійснювати облік прийнятих, переміщених та звільнених працівників.

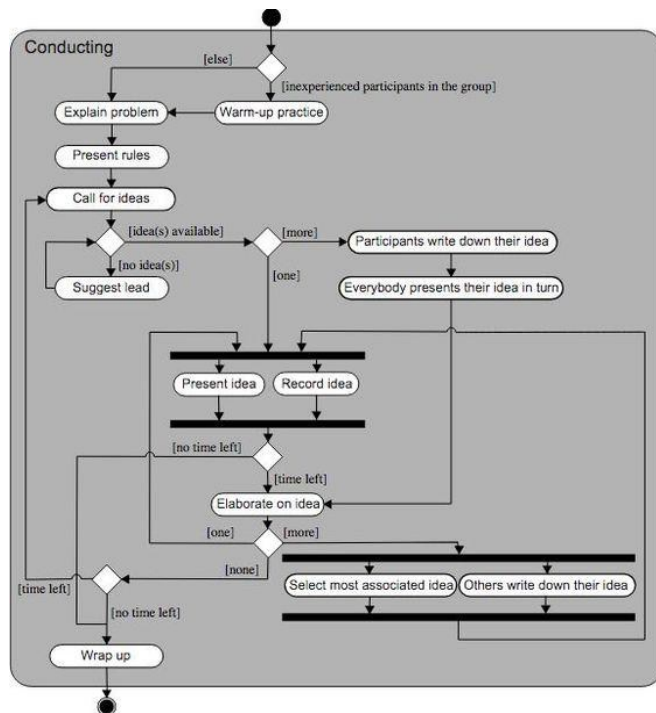
На основі цієї інформації проводити аналіз кадрового стану підприємства, а також формувати відомості прийнятих, переміщених та звільнених працівників за період з можливістю їх роздрукування.

Завдання:

- побудувати діаграму варіантів використання для заданої предметної області;
- описати сценарії двох варіантів використання із числа тих, що наведені на діаграмі варіантів використання
- побудувати діаграму діяльності,
- побудувати діаграму класів.

Завдання 2

На наведеній діаграмі вказати тип діаграми, назви структурних елементів і позначень, детально описати, що саме відображає діаграма.



До файлу звіту включити: 1. побудовані діаграми: Use-case, activity та класів; таблицю з описом сценаріїв варіантів використання; 2. опис діаграми та її структурних елементів.

Затверджено на засіданні кафедри інформаційних систем
 протокол № ____ від «__» _____ 20__ р.

Екзаменатор

к.е.н., проф. Золотарьова І.О.

Зав. кафедрою

к.т.н., доц. Бондаренко Д.О.

Критерії оцінювання

Підсумкові бали за екзамен складаються із суми балів за виконання всіх завдань, що округлені до цілого числа за правилами математики.

Алгоритм вирішення кожного завдання включає окремі етапи, які відрізняються за складністю, трудомісткістю та значенням для розв'язання завдання. Тому окремі завдання та етапи їх розв'язання оцінюються відокремлено один від одного таким чином:

Перше завдання оцінюється від 0 до 30 балів відповідно до наступних складових:

10 балів – правильна та повна побудова діаграми варіантів використання та правильний та повний опис сценаріїв двох варіантів використання;

10 балів – правильна та повна побудова діаграми класів;

10 балів - правильна та повна побудова діаграми діяльності.

У випадку, коли описані вище частини завдання виконані не повністю, з максимального балу знімається по 1 балу.

Друге завдання оцінюється від 0 до 10 балів відповідно до наступних складових:

10 балів – правильний та повний опис складових наведеної діаграми. У випадку, коли частини завдання виконані не повністю, з максимального балу знімається по 1 балу.

РЕКОМЕНДОВАНА ЛІТЕРАТУРА

Основна

1. Fundamental Approaches to Software Engineering. 22nd International Conference, FASE 2019, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2019, Prague, Czech Republic, April 6–11, 2019, Proceedings
2. UML-Based Software Product Line Engineering with Smarty. Published by Springer International Publishing, 2023, <https://doi.org/10.1007/978-3-031-18556-4>
3. І. Бородкіна, Г. Бородкін, Інженерія програмного забезпечення. Посібник для студентів вищих навчальних закладів – К., «Центр навчальної літератури» – 2018.–204 с.
4. Software Engineering at Google: Lessons Learned from Programming Over Time by Titus Winters (Author), Tom Manshreck (Author), Hyrum Wright (Author). O'Reilly Media; 1st edition (April 7, 2020).
5. Modern Software Engineering. Doing What Works to Build Better Software Faster. David Farley. 2021. Addison-Wesley Professional.
6. Максвел К. Джон П'ять рівнів лідерства.: Пер. з англ. Т. Куріпко. – Х.: Вид-во «Ранок» : «Фабула», 2019. – 304 с.
7. Джозеф Хігні. Основи управління проєктами. Фабула. - 2020.- 272с.
8. Gregory M. Horine. Project Management Absolute Beginner's Guide 5th Edition, 2022 <https://a.co/d/7oJCrpT>
9. Річард Ньютон. Управління проєктами від А до Я. Альпіна Паблішер, - 2018. - 180 с.
10. 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.
11. Мартин Фаулер. UML. Основы. 3-е издание / Мартін Фаулер, Кендалл Скотт – М.: Символ-плюс, 2018. – 192 с.
12. Роберт М. Чиста архітектура: мистецтво розробки програмного забезпечення» / Роберт Мартін, Фабула, 2019. – 416 с.
13. Сучасні інформаційні технології та системи [Електронний ресурс] : монографія / Н. Г. Аксак, Л. Е. Гризун, О. В. Щербаков [та ін.] ; за заг. ред. Пономаренка В. С. — Електрон. текстові дан. (22,9 МБ). — Харків : ХНЕУ ім. С. Кузнеця, 2022. — 270 с. — Режим доступу: <http://www.repository.hneu.edu.ua/handle/123456789/29233>.
14. Yuriy Skorin, Iryna Zolotaryova. Enhancing the effectiveness of usability testing for user interfaces / // International scientific journal “Computer systems and information technologies”. № 3. – Khmelnytskyi : Khmelnytskyi National University, 2023. – С. 65–74. – Режим доступу: <http://www.repository.hneu.edu.ua/handle/123456789/30712>.
15. SWEEBOK (Software Engineering Body of Knowledge) <https://www.computer.org/education/bodies-of-knowledge/software-engineering>.

16. ISO/IEC/IEEE 90003:2018 Software engineering
<https://www.iso.org/standard/74348.html>
17. ISO/IEC/IEEE 12207:2017 Systems and software engineering Software life cycle processes <https://www.iso.org/standard/63712.html>

Інформаційні ресурси

Інструменти та стандарти управління вимогами :

- [Doors Next](#)
- [ReqView](#)
- [Visure Requirements Management](#)
- [ISO/IEC/IEEE International Standards](#)
- [Volere Process](#)
- [Mastering the Requirements Process](#)

Інструменти UML:

- [UML Resource Center at IBM Rational](#)
- Agile Modeling (<http://www.agilemodeling.com/>), also refer to Extreme Programming (<http://www.extremeprogramming.org/>).
- <http://www.objectsbydesign.com/>
- [A nice page on how to choose a UML modeling tool](#)
- OMG's UML resource page: <http://www.omg.org/uml/>
- [Unified Modeling Language \(UML\)](#)
- [Common Warehouse Metamodel \(CWM\)](#)
- [IBM Technical Journals](#)
- <http://www.holub.com/goodies/uml/index.html>
- <http://www.smartdraw.com/resources/centers/uml/uml.htm>
- <https://github.com/Alliedium/awesome-software-engineering?tab=readme-ov-file>