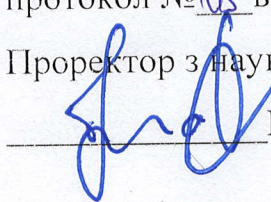


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
імені ІВАНА ПУЛЮЯ

УХВАЛЕНО

Науково-технічною радою ТНТУ
протокол № 05 від 02.04 2025 р

Проректор з наукової роботи
 Павло МАРУЩАК

ЗАТВЕРДЖЕНО



Голова приймальної комісії

Микола МИТНИК

» 02.04 2025 р

ПРОГРАМА

вступу на навчання для здобуття ступеня доктора філософії

Галузь знань F «Інформаційні технології»

Спеціальність: F2 «Інженерія програмного забезпечення»

Освітньо-наукова програма Інженерія програмного забезпечення

ТЕРНОПІЛЬ – 2025

ЗМІСТ

- I. ЗАГАЛЬНІ ПОЛОЖЕННЯ
- II. ПОРЯДОК ПРОВЕДЕННЯ СПІВБЕСІДИ
- III. КРИТЕРІЇ ОЦІНЮВАННЯ СПІВБЕСІДИ
- IV. ЗМІСТ ПРОГРАМИ ВСТУПНОГО ІСПИТУ
- V. ПОРЯДОК ПРОВЕДЕННЯ ТА КРИТЕРІЇ ОЦІНЮВАННЯ ВСТУПНОГО ІСПИТУ
- VI. ВИМОГИ ДО НАПИСАННЯ ДОСЛІДНИЦЬКОЇ ПРОПОЗИЦІЇ
- VII. ОСКАРЖЕННЯ РЕЗУЛЬТАТІВ ВСТУПНИХ ВИПРОБУВАНЬ
- ДОДАТКИ

I. ЗАГАЛЬНІ ПОЛОЖЕННЯ

Програма вступу на навчання для здобуття ступеня доктора філософії складена відповідно до Порядку підготовки здобувачів вищої освіти ступеня доктора філософії та доктора наук у вищих навчальних закладах (наукових установах), затвердженого постановою Кабінету Міністрів України від 23 березня 2016 р № 261, Положення про підготовку здобувачів вищої освіти ступеня доктора філософії та доктора наук у Тернопільському національному технічному університеті імені Івана Пулюя <https://docs.tntu.edu.ua/base/document?id=1112> та Правил прийому до аспірантури Тернопільського національного технічного університету імені Івана Пулюя у <https://tntu.edu.ua/storage/pages/00000314/tntu-pp2025-dodatky.pdf>.

Вступ на навчання до аспірантури здійснюється за результатами співбесіди та вступного іспиту зі спеціальності.

II. ПОРЯДОК ПРОВЕДЕННЯ СПІВБЕСІДИ

Метою співбесіди є оцінювання спроможності вступників застосовувати набуті знання у науковому дослідженні за обраною спеціальністю. Результати співбесіди є підставою для допуску до наступного етапу вступного випробування.

Співбесіда проводиться в усній формі. На співбесіді відбувається презентація дослідницької пропозиції чи розгляд наукових праць (не менше однієї фахової статті категорії Б чи статті у періодичних наукових виданнях, проіндексованих у базах даних Web of Science Core Collection та/або Scopus, крім видань держави, визнаної Верховною Радою України державою-агресором) та не менше двох тез доповідей, які відповідають обраній спеціальності. Результати співбесіди зазначаються у протоколі співбесіди (додаток Б).

Попередньо члени предметної комісії ознайомлюються із мотиваційними листами вступників. Оцінюються мотиваційні листи лише щодо тих вступників, які мають однаковий конкурсний бал у рейтинговому списку (додаток 7 до Правил прийому на навчання до Тернопільського національного технічного університету імені Івана Пулюя в 2025 році).

За результатами співбесіди, розгляду поданих наукових праць щодо їх відповідності обраному напрямку дослідження або розгляду поданого наукового тексту предметна комісія приймає рішення про те, що вступника «рекомендовано» або «не рекомендовано» до участі в наступних вступних випробуваннях.

Особи, які без поважних причин не з'явилися на співбесіду у визначений час та особи щодо яких комісія прийняла рішення «не рекомендовано» до вступного іспиту зі спеціальності не допускаються і не беруть участь у конкурсному відборі.

Дослідницька пропозиція та/або копії наукових публікацій вступника надаються предметній комісії, яка проводить співбесіду.

Під час проходження співбесіди вступник може:

- поділитися попереднім науковим досвідом (наявності наукових публікацій, участі у наукових заходах та наукових проєктах, наукових конкурсах, наукових школах, роботи в наукових гуртках);
- висловлюватися з приводу бажаного напрямку наукових досліджень, наукової спеціальності, ймовірного наукового керівника, можливості участі в програмах академічної мобільності, стажуваннях, у тому числі за кордоном тощо;
- користуватися копіями наукових публікацій для підтвердження тих чи інших висловлених положень, обчислень та цитування думок інших вчених, на висновки яких посилається автор під час проведеного дослідження.

Під час проходження співбесіди вступник повинен:

- бути спроможним пояснити та аргументувати наукові результати, отримані автором під час попереднього наукового досвіду, викладеного у наукових публікаціях (у тому числі визначення особистого внеску в публікаціях, виконаних у співавторстві), чи наукової розвідки, здійсненої у вигляді дослідницької пропозиції (актуальність обраної тематики, ступінь її розробленості у вітчизняній і зарубіжній науці, наявність власних оригінальних висновків щодо наявних положень чи концепцій, які стосуються предмету дослідження тощо);
- представити бачення проведення майбутнього наукового дослідження, його проміжних та кінцевих результатів, а також можливості їхнього впровадження у різні сфери людської діяльності за спеціальністю;
- продемонструвати розуміння засад наукової-дослідної роботи (основ культури мовлення, академічного письма, етики наукової діяльності, методології наукових досліджень).

За потреби предметна комісія може ставити уточнюючі питання.

ІІІ. КРИТЕРІЇ ОЦІНЮВАННЯ СПІВБЕСІДИ

При визначенні результатів співбесіди та для прийняття рішення про рекомендацію або не рекомендацію до подальшого проходження вступником до аспірантури вступних випробувань предметна комісія враховує:

- уміння обґрунтувати свою думку;
- уміння виявляти теоретичні та практичні проблеми певної наукової сфери;
- здатність формулювати завдання задля поставлених цілей наукового дослідження;

- уміння аналізувати та систематизувати наукові джерела та інформацію з них, розуміння основних наукових концепцій, які існують за обраним напрямом наукового дослідження;

- вміння коректно, стисло, точно відповідати на запитання.

Рішення про те, щоб вступника НЕ рекомендувати до подальшого проходження вступних випробувань предметною комісією може бути прийняте у разі, якщо:

1. Вступник не може обґрунтувати актуальність, об'єкт, завдання, предмет та методи дослідження, відображені у наукових публікаціях (дослідницькій пропозиції);

2. Відповіді на уточнюючі питання не відображають розуміння основ ведення науково-дослідної роботи;

3. Більшість відповідей на запитання неточні або неправильні;

4. Існують обґрунтовані сумніви щодо самостійного виконання дослідницької пропозиції, що підтверджено низкою питань особи, яка проводить співбесіду, із фіксуванням у протоколі співбесіди.

I V. ЗМІСТ ПРОГРАМИ ВСТУПНОГО ІСПИТУ ЗІ СПЕЦІАЛЬНОСТІ F2 «ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»

Метою складання вступного іспиту зі спеціальності є перевірка й оцінювання знань вступників із основних дисциплін, що становлять ядро інженерії програмного забезпечення.

Програму вступного іспиту складено в обсязі програм попередніх рівнів вищої освіти зі спеціальності F2 «Інженерія програмного забезпечення» що відображає кваліфікаційні вимоги до теоретичних знань та практичних навичок вступників до аспірантури, а також свідчить про їх компетентну здатність застосовувати набуті вміння при проведенні наукових досліджень. Основу програми склали ключові положення таких дисциплін як: «Алгоритми та структури даних», «Архітектура та проектування програмного забезпечення, управління ІТ-проектами», «Математичне моделювання процесів та об'єктів предметних областей», «Моделювання складних розподілених систем та об'єктів», «Моделювання та архітектура програмного забезпечення», «Об'єктно-орієнтоване програмування та об'єктне моделювання програмного забезпечення на основі OCL».

На основі яких складається перелік питань вступного іспиту й формуються екзаменаційні білети.

1. АЛГОРИТМИ ТА СТРУКТУРИ ДАНИХ

Зміст дисципліни:

Зміст дисципліни “Алгоритми і структури даних” полягає в тому, щоб навчити здобувачів основним відомим структурам даних та способах їх реалізації при допомозі

мов програмування, найбільш важливим базовим алгоритмам, що застосовуються на сьогодні, а також принципам та методам побудови алгоритмів та їх оцінки.

Необхідно знати: що таке алгоритм, способи його представлення та основні властивості; що таке абстрактні типи даних; базові абстракти даних: дерево та способи їх реалізації; основні алгоритми пошуку та сортування; принципи побудови алгоритмів та методи оцінювання їх ефективності.

Необхідно вміти: представляти алгоритми при допомозі алгоритмічної мови та блок-схем; реалізовувати основні типи даних; реалізовувати структури даних типу дерево, множина, орієнтований граф, неорієнтований граф. реалізовувати основні алгоритми пошуку; реалізовувати прості алгоритми сортування; реалізовувати алгоритми швидкого пірамідального та “кишенькового” сортування.

Тематика завдань:

1. Поняття алгоритму. Означення алгоритму. Приклади алгоритмів
2. Основні властивості алгоритмів.
3. Способи запису алгоритмів
4. Етапи повної побудови алгоритму
5. Часова складність алгоритму
6. Асимптотичні співвідношення. O-символіка.
7. Основні правила аналізу часу виконання алгоритмів
8. Аналіз об'єму пам'яті, що використовуватиметься програмною
9. Поняття абстрактного типу даних (АТД)
10. Тип даних “Список”, його властивості. Оператори для роботи із АТД “Список”
11. Реалізація АТД “Список” при допомозі масивів
12. Реалізація АТД “Список” при допомозі вказівників. Однозв'язні списки
13. Реалізація АТД “Список” при допомозі вказівників. Двозв'язні списки
14. Реалізація АТД “Список” при допомозі курсорів
15. Тип даних “Стек”, його властивості. Оператори для роботи із АТД “Стек”
16. Реалізація АТД “Стек” при допомозі масиву
17. Тип даних “Черга”, його властивості. Оператори для роботи з АТД “Черга”
18. Реалізація АТД “Черга” при допомозі вказівників
19. Реалізація АТД “Черга” при допомозі циклічного масиву
20. Тип даних “Відображення”, його властивості. Оператори для роботи із АТД “Відображення”
21. Реалізація АТД “Відображення” при допомозі масиву
22. Реалізація АТД “Відображення” при допомозі списків
23. Тип даних “Множина”. Основні дії над множинами

24. Оператори для роботи із АТД “Множина”. Моделі представлення.
25. Реалізація АТД “Множина” при допомозі двійкових векторів
26. Реалізація АТД “Множина” при допомозі зв’язаних списків
27. АТД “Словник”. Реалізація словників
28. Хешування та хеш-таблиці. Розв’язання колізій при хешуванні.
29. Відкрите та закрите хешування
30. Хеш-функція. Приклади хеш-функцій
31. Тип даних “Дерево”. Основні означення. Оператори для роботи із АТД “Дерево”. Моделі реалізації АТД
32. Алгоритми обходу дерева в прямому, зворотному та симетричному порядку
33. Дерева з мітками. Дерева-вирази.
34. Реалізація АТД “Дерево” при допомозі масиву
35. Реалізація АТД “Дерево” при допомозі списків синів
36. Бінарні дерева.
37. Задача побудови коду Хаффмана.
38. Частково-впорядковане дерево.
39. Поняття орієнтованого графу. Операції, що виконуються на графі
40. Структури даних для представлення орієнтованих графів
41. Вирішення задачі відшукування найкоротшого шляху з використанням орграфу
42. Алгоритм знаходження найкоротших шляхів між двома парами вершин орграфу
43. Алгоритм знаходження існування шляхів між вершинами орграфу
44. Алгоритм обходу орієнтованого графу. Пошук в глибину.
45. Неорієнтовані графи та способи їх представлення
46. Охоплюючи дерево та алгоритми його побудови
47. Алгоритм пошуку в ширину на графі
48. Постановка задачі сортування. Види сортування. Прості алгоритми сортування
49. Алгоритм швидкого сортування
50. Алгоритм пірамідального сортування
51. Алгоритм коміркового сортування
52. Алгоритм сортування злиттям
53. Алгоритм сортування Шелла
54. Алгоритм сортування за розрядами
55. Постановка задачі пошуку. Оптимальність пошуку. Алгоритми пошуку.
56. Алгоритм пошуку з використання індексації по ключах
57. Алгоритм послідовного пошуку для впорядкованих та неупорядкованих

Масивів.

58. Алгоритм бінарного пошуку.

59. Двійкове дерево пошуку. Балансування дерева

60. Методи розробки алгоритмів

61. Алгоритми класу “розділяй і володарюй”

62. Алгоритми динамічного програмування

63. «Жадібні» алгоритми

64. Пошук з поверненням. Функція виграшу та альфа-бета відсікання

65. Алгоритм локального пошуку. Розв’язання задачі комівояжера

Рекомендована література:

1. Data Structures and Algorithms Alfred V. Aho , John E. Hopcroft , Jeffrey D. Ullman – Pearson, 2010. - 448p.
2. Introduction to Algorithms Thomas H. Cormen , Charles E. Leiserson , Ronald L. Rivest , Clifford Stein – MIT Press, 2009. - 1312p.
3. Knuth D. The Art of Computer Programming, Volumes 1-4a. Donald Ervin Knuth - Addison-Wesley Professional, 2011. – 3168p
4. Sedgewick R. Algorithms Robert Sedgewick , Kevin Wayne - Addison-Wesley Professional, 2011. – 976p
5. Algorithms in C++, Parts 1-4: Fundamentals, Data Structure, Sorting, Searching, Third Edition Robert Sedgewick - Addison-Wesley Professional, 1998. – 738p

2. АРХІТЕКТУРА ТА ПРОЕКТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ, УПРАВЛІННЯ ІТ-ПРОЕКТАМИ

Зміст дисципліни:

Зміст дисципліни “Архітектура та проектування програмного забезпечення” полягає в здобутті студентами теоретичних знань щодо принципів, технологій, підходів, методів та засобів проектування архітектури програмних систем, а також практичних навичок щодо виконання відповідних фаз створення програмних проектів та продуктів.

Завданням вивчення навчальної дисципліни є надання знань та компетенцій щодо сучасних архітектур програмних систем, моделей життєвого циклу програмних систем, методів, засобів та технологій розробки архітектури складних програмних систем; вміння розв’язувати задачі, що виникають на окремих фазах життєвого циклу програмних систем та пов’язані із проектування архітектури програмних систем; володіти сучасними засобами проектування інформаційних систем, патернами проектування та архітектурними шаблонами. Задача формування у студентів комплексу знань та практичних навичок із планування, моніторингу та управління проектами від локального до корпоративного рівнів. Розглядаються принципи управління проектами розробки інформаційних систем, структурної організації

команди розробників системи, їх взаємодія в процесі функціонування, перспективні напрямки розвитку сучасних інформаційних технологій

Тематика завдань:

1. Цілі (goals) і завдання проектування
2. Значимість конспекту (середовища розроблення та застосування) для розуміння ролі проектування програмного забезпечення
3. Особливості та вплив контексту, в якому здійснюється проектування і використовуються його результати
4. Життєвий цикл програмної інженерії як контекст проектування.
5. Зв'язок проектування з результатами аналізу вимог, конструюванням програмних систем та їх тестуванням.
6. Стандарти життєвого циклу ПЗ (IEEE и ISO/IEC). Спеціальні аспекти проєктирования, їх деталізація.
7. Опис контексту проєктирования: від вимог до тестів.
8. Двокроковий процес проектування. Архітектурне проєктирование - декомпозиція структури (*статична*) і організації (*динамічна*) компоненти.
9. Деталізація архітектури (описує специфічну поведінку и характеристики окремих компонентів).
10. Результати процесу проектування: набір моделей і артефактів, що містять прийнятих рішень згідно способів реалізації вимог у програмному коді.
11. Основні принципи проектування (Enabling Techniques) як ключові фундаментальні
12. Ідеї і концепції в різних методах і подходах проектування ПЗ.
13. Що таке абстракція (Abstraction).
14. Два механізми абстракції в контексті проектування програмних систем.
15. Параметризація і специфікування (деталізація).
16. Три види абстракції через специфікування.
17. Процедурна абстракція (динамічна, у відношенні поведінки).
18. Статична абстракція даних (у відношенні інформації)
19. Абстракція контролю (управління системою і інформацію, що обробляється нею).
20. Зв'язність і з'єднання (Coupling and Cohesion).
21. Зв'язність (*Coupling*) як сила зв'язку чи взаємовпливу (ступінь його самодостатності або відсутності) між модулями.
22. З'єднання (*Cohesion*) як механізм забезпечення внутрішнього зв'язку
23. Декомпозиція і розбиття на модулі (Decomposition and Modularization):
24. Способи отримання дрібніших відносно незалежних програмних компонент, кожен з яких несе різну функціональність (логічно зв'язані групи функціональності).
25. Інкапсуляція/приховування інформації (Encapsulation/information hiding).
26. Концепція групування і упакування елементів в внутрішніх деталях абстракції (моделі) по відношенню до реалізації.

27. Умови доступу спадкоємці компонентів до внутрішніх деталей реалізації компоненти, що є їх предком.
28. Взаємовідносини: Користувач – Компонента
29. Розділення інтерфейса і реалізації (Separation of interface and implementation).
30. Визначення компоненти через специфікування інтерфейсу і доступного клієнтам (чи іншим компонентам) від безпосередніх деталей реалізації.
31. Достатність, повнота і простота (Sufficiency, completeness and primitiveness).
32. Достатність і функціональність, що відсутня в моделі.
33. Співвідношення ключових проблем проектування (Key Issues in Software Design) і способів проектування: проведення декомпозиції; об'єднання компонентів єдиної системи; забезпечення необхідної продуктивності та якості системи
34. Паралелізм (Concurrency): підходи і методи організації процесів, задач і потоків для забезпечення ефективності, атомарності, синхронізації і розподілу в часі обробки інформації.
35. Контроль і обробка подій (Control and Handling of Events).
36. Неявні методи обробки подій, що реалізовані часто у вигляді функції зворотного виклику (call-back), як однієї з фундаментальних концепцій обробки подій.
37. Розподіл і виконання компонентів (Distribution of Components) по різних вузлах обробки в термінах апаратного забезпечення, мережевої інфраструктури
38. Використання зв'язуючого програмного забезпечення (middleware)
39. Обробка помилок та виняткових ситуацій і забезпечення відмовостійкості (Errors and Exception Handling and Fault Tolerance).
40. Попередження збоїв та забезпечення подальшого функціонування системи, якщо збої настануть.
41. Взаємодія і представлення (Interaction and Presentation).
42. Представлення інформації користувачам та взаємодії користувача з системою з точки зору реакції системи на дії користувачів.
43. Реакція системи у відповідь на дії користувачів і організація її відклику з точки зору внутрішньої організації взаємодії.
44. Збереженість даних (Data Persistence).
45. Концепції Model-View-Controller.
46. Проблеми обробки “довгоживучих” даних.
47. Суть архітектури ПЗ
48. Архітектурно-орієнтований процес проектування ПЗ
49. Як отримати архітектуру
50. Як описати архітектуру
51. Зразки архітектури
52. Зв'язок ВВ і архітектури
53. Ризики, пов'язані з архітектурою
54. Що є базовим рівнем архітектури ПЗ
55. Опис архітектури
56. Суть архітектурного представлення моделі ВВ

57. Суть архітектурного представлення моделі проектування
58. Суть архітектурного представлення моделі реалізації
59. Що ми розуміємо під архітектурою програмної системи
60. Для чого потрібна архітектура ПЗ (розуміння системи, організація розробки, повторне використання розвиток системи)
61. Кроки розробки архітектури
62. Способи отримання стійкої архітектури
63. Сфери відповідальності архітектора в проектуванні ПЗ
64. Процес проектування, керований варіантами використання
65. Ітеративний та інкрементний процес
66. Процес проектування, керований варіантами використання, архітектурно-орієнтований, ітеративний та інкрементний
67. Уніфікований процес проектування ПЗ.
68. Ітерації, їх роль
69. Що не включається в ітерацію?
70. Для чого використовується ітеративна та інкрементна розробка
71. За рахунок чого можна забезпечити зниження ризиків
72. Отримання стійкої архітектури
73. За рахунок чого досягається підтримка вимог, що змінюються
74. Отримання доступності тактичних змін
75. Досягнення постійної цілісності ПЗ
76. Досягнення легкості і зручності навчання персоналу
77. Що є потенційними ризиками розроблення ПЗ
78. Що є результатом ітерації
79. Життєвий цикл (ЖЦ) ПЗ
80. Роль вимог в ЖЦ ПЗ
81. Ітерації в ЖЦ ПЗ
82. Мета процесу визначення вимог
83. Суть процесу визначення вимог
84. що таке контекст системи
85. Модель предметної області системи
86. Суть розробки моделі предметної області системи
87. Що таке бізнес-модель системи
88. Як розробити бізнес-модель
89. Зв'язок контексту системи з бізнес-моделлю
90. Чотири «П» - складові розробки ПЗ
91. Основні робочі процеси розроблення (*визначення вимог, аналіз, проектування, реалізація, тестування*)
92. Узагальнений робочий процес ітерації
93. Процес як шаблон проектування
94. Робочий процес: артефакти, співробітники, діяльності
95. Визначення вимог у вигляді ВВ
96. Що рухає проектом
97. Завдання UML

- 98. Означення варіантів використання (прецедентів)
- 99. Ідентифікація ВВ
- 100. Опис ВВ
- 101. Опис моделі ВВ
- 102. Що таке Use case діаграма
- 103. Діаграма станів для ВВ
- 104. Діаграми взаємодій
- 105. Що таке діаграма класів і розширена діаграма класів
- 106. Що відображає модель варіантів використання
- 107. Суть актантів
- 108. Категорії актантів. Зовнішні і внутрішні актантів
- 109. Задачі ідентифікації актантів
- 110. Що таке артефакти
- 111. Що є артефактами проектування ПЗ
- 112. Що визначають варіанти використання
- 113. Для чого вимоги оформляють у вигляді варіантів використання
- 114. Що таке співробітники
- 115. Співробітник: Системний аналітик, його роль
- 116. Співробітник: Архітектор, його роль
- 117. Співробітник: Специфікатор ВВ, його роль
- 118. Співробітник: Розробник інтерфейса користувача, його роль
- 119. Діяльності робочого процесу, їх види
- 120. Суть аналітичної моделі
- 121. Що таке кооперації класів і яка їх роль в проектуванні
- 122. В чому суть відповідальності класів
- 123. Діаграма кооперації реалізації ВВ
- 124. Які можуть існувати відношення між класами
- 125. Граничні класи
- 126. Класи сутностей
- 127. Класи управління
- 128. Що таке інтерфейс
- 129. Що таке реалізація
- 130. Асоціації
- 131. Діаграми об'єктів (послідовностей і дій)
- 132. Заступники об'єктів
- 133. Артефакт: Модель проектування
- 134. Вихідні дані і результат проектування
- 135. Ієрархія підсистем проектування
- 136. Артефакт: Клас проектування
- 137. Визначення методів класу проектування
- 138. Визначення атрибутів класу проектування
- 139. Визначення асоціацій і агрегацій
- 140. Визначення архітектурно-значимих класів проектування
- 141. Активні класи

- 142. Ключові атрибути асоціації і класу проектування
- 143. Артефакт: Проект реалізації ВВ
- 144. Артефакт: Інтерфейс
- 145. Артефакт: Опис архітектури (представлення моделі проектування)
- 146. Артефакт: Модель розгортання
- 147. Діяльність : Проектування архітектури
- 148. Діяльність : Проектування ВВ
- 149. Діяльність : Проектування Класу
- 150. Діяльність : Проектування Підсистеми
- 151. Зв'язок підсистем аналізу і проектування
- 152. Складові підсистем
- 153. Об'єктна бізнес-модель
- 154. Описи взаємодій об'єктів проектування
- 155. Суть бізнес-моделювання для проектування ПЗ
- 156. Сутність трасування моделей
- 157. Роль реалізації в ЖЦ ПЗ
- 158. Артефакт: Модель реалізації
- 159. Ієрархія підсистем реалізації
- 160. Артефакт: Компонента
- 161. Артефакт: Підсистема реалізації
- 162. Зв'язок підсистем проектування і реалізації
- 163. Артефакт: Інтерфейс
- 164. Задача створення прототипу інтерфейса користувача
- 165. Діяльність : Реалізація архітектури
- 166. Діяльність : Реалізація підсистеми
- 167. Діяльність : Реалізація класу
- 168. Діяльність: Компонування системи
- 169. Суть архітектурного представлення моделі реалізації
- 170. Суть тестування ВВ
- 171. Артефакт: Модель тестування
- 172. Артефакт: План тестування
- 173. Артефакт: Оцінка тесту
- 174. Діяльність : Планування тестування
- 175. Діяльність : Розробка тесту
- 176. Діяльність : Проведення тестування цілісності системи
- 177. Діяльність : Оцінка результатів тестування
- 178. Узагальнений робочий процес ітерації
- 179. Фази ЖЦ, на які попередньо розбивається процес розробки ПЗ (*чотири фази*)
- 180. Взаємовідношення між фазами, ітераціями та робочими процесами (*5 РП*)
- 181. Фаза аналізу і планування вимог (*визначає виконуваність проекту*)
- 182. Цілі фази аналізу і планування вимог
- 183. Пакети аналізу
- 184. Фаза проектування (*забезпечує можливість виконання*)

185. Цілі Фази проектування
186. Фаза побудови (*створює систему*)
187. Цілі фази побудови
188. Фаза впровадження (*переносить систему в середовище користувачів*)
189. Цілі фази впровадження
190. Суть узагальненої ітерації
191. Вплив ризиків на план ітерацій і планування проекту
192. Чому основні робочі процеси (В, А, П, Р, Т) повторюються на кожній ітерації
193. Роль планування ітерації
194. Планування критеріїв оцінки
195. Основні критерії оцінки проектів (*складний проект вимагає великого, нова лінія вимагає досвіду, ціна за використання ресурсів*)
196. Оцінка ітерацій і фаз
197. Які основні потоки робіт ітерації здійснюються на фазі аналізу і планування вимог
198. Чому фаза проектування забезпечує створення базового рівня архітектури
199. Чому універсальний процес допомагає справитися зі складністю
200. Планування фази аналізу і планування вимог
201. Виконання основних робочих процесів від визначення вимог до тестування
202. Визначення ітерацій у фазі аналізу і планування вимог
203. Визначальна фаза для створення базового рівня архітектури

Рекомендована література:

1. Documenting Software Architectures: Views and Beyond (2nd Edition)
2. Андрій Будай. Дизайн-патерни — просто, як двері
3. Arlow J. UML 2 & the Unified Process: Practical ObjectOriented Analysis and Design, 2nd Ed. / Jim Arlow, Ila Neustadt - Addison_Wesley, 2005.
4. Rumbaugh J. The Unified Modeling Language Reference Manual», 2nd Ed. / James Rumbaugh, Ivar Jacobson, Grady Booch - Addison_Wesley, 2005.
5. Booch G. Object-Oriented Analysis and Design with Applications, 3 Ed. / Grady Booch - Addison-Wesley, 2007.
6. UML 2.0 Object Constraint Language (OCL), www.omg.org. (Об'єктна мова обмежень – текстова мова обмежень і запитів), OMG (Object Management Group). www.uml.org (2004)
7. Cours de Jean-Marie Favre. Paris, IMAG, 2014.
8. Петрик М. Моделювання та аналіз програмного забезпечення / Петрик М., Петрик О. -Тернопіль: Видавництво ТНТУ, 2015. – 210 с.
9. UML 2.0 et UML 2.0 Infrastructure /Benoît Charoux, Aomar Osmani, Yann Thierry-Mieg. Editions Pearson, Education France, 2008.
10. Alexander I. Writing Better Requirements/ Ian F. Alexander, Richard Stevens, Ian Alexander, Addison_Wesley - 2002, ISBN 0321131630.

11. Ambler S.W. The Unified Process Inception Phase / Scott W. Ambler, Larry L. Constantine - CMP Books, 2000, ISBN 1929629109.
12. Ambler S.W. The Unified Process Elaboration Phase / Scott W. Ambler, Larry L. Constantine, Roger Smith - CMP Books, 2000, ISBN 1929629052.
13. Ambler S.W. The Unified Process Construction Phase / Scott W. Ambler, Larry L. Constantine - CMP Books, 2000, ISBN 192962901X.
14. Arlow J. Enterprise Patterns and MDA: Building Better Software with Archetype Patterns and UML», Jim Arlow, Ila Neustadt, Addison Wesley - 2004, ISBN 032111230X.
15. Frankel D.S. Model Driven Architecture: Applying MDA to Enterprise Computing/ David S. Frankel, John Wiley & Sons, 2003, ISBN 0471319201.
16. Kay M. XSLT Programmer's Reference. 2nd Edition / Michael Kay - Wrox Press Inc., 2001, ISBN 1861005067.
17. Kleppe A. MDA Explained: The Model Driven Architecture – Practice and Promise / Anneke Kleppe, Jos Warmer, Wim Bast - Addison_Wesley, 2001, ISBN 032119442X.
18. Kroll P. The Rational Unified Process Made Easy: A Practitioner's Guide to Rational Unified Process / Per Kroll, Philippe Kruchten, Grady Booch - Addison_Wesley, 2003, ISBN 0321166094.
19. Kruchten P. The Rational Unified Process, An Introduction/ Philippe Kruchten - Addison_Wesley, 2000, ISBN 0201707101.
20. Kruchten P. The 4+1 View of Architecture / Philippe Kruchten, IEEE Software, 12(6) Nov. 1995, pp. 45–50.
21. Leffingwell D. Managing Software Requirements: A Use Case Approach, 2nd Edition / Dean Leffingwell, Don Widrig - Addison_Wesley, 2003, ISBN 032112247X.
22. Meyer B. Object Oriented Software Construction / Bertrand Meyer, Prentice Hall, 1997, ISBN 0136291554.
23. Mellor S. Agile MDA / Stephen J Mellor, MDA Journal», June 2004. [Електронний ресурс] – Режим доступу: www.bptrends.com.
24. Unified Modeling Language: OCL, version 2.0 . [Електронний ресурс] – Режим доступу: www.omg.org.
25. Pitts N. XML Black Book, 2nd Edition // Natalie Pitts, Coriolis Group, 2000, ISBN 1576107833.
26. Rumbaugh J. The Unified Modeling Language Reference Manual, 2nd Edition/ James Rumbaugh, Ivar Jacobson, Grady Booch - Addison Wesley, 2004, ISBN 0321245628.
27. Unified Modeling Language: Superstructure, version 2.0. . [Електронний ресурс] – Режим доступу: www.omg.org.
28. Warmer J. The Object Constraint Language: Getting Your Models Ready for MDA / Jos Warmer, Anneke Kleppe - Addison_Wesley, 2003, ISBN 0321179366.

3. МАТЕМАТИЧНЕ МОДЕЛЮВАННЯ ПРОЦЕСІВ ТА ОБ'ЄКТІВ ПРЕДМЕТНИХ ОБЛАСТЕЙ

Зміст дисципліни:

Метою дисципліни "Математичне моделювання процесів та об'єктів предметних областей" є ознайомлення студентів з основами вибраного ними професійного напрямку підготовки та надання їм базових знань із ключових теоретичних і практичних понять та основних перспективних напрямків математичного моделювання, що потребують застосування методів інженерії програмного забезпечення. Сформувані основи інструментальних засобів побудови розв'язків математичних моделей прикладного характеру, що описуються диференціальними рівняннями.

Завданням дисципліни є формування прикладного математичного фундаменту бакалавра комп'ютерних технологій, що виражається у спроможності застосовувати і розвивати отримані знання для вирішення проблем наукових досліджень, проектування та експлуатації систем обробки інформації та прийняття рішень, задач, що потребують використання сучасних методів математичного моделювання та інформаційних технологій.

Тематика завдань:

1. Інженерно-технологічні та фізичні задачі, що приводять до диференціальних рівнянь. Основні використовувані визначення теорії диференціальних рівнянь. Фізичний зміст похідної та диференціалу. Однорідні диференціальні рівняння. Однорідні та неоднорідні диференціальні рівняння, рівняння, що зводяться до них. Приклади описання фізичних процесів за допомогою математичних моделей. Принципи побудови математичних моделей. Поняття адекватності математичних моделей.
2. Методика розв'язання найпростіших обернених задач з використанням результатів експериментальних спостережень (слідів розв'язків). Використання математичних пакетів для розв'язання диференціальних рівнянь. Рівняння в повних диференціалах. Диференціальні рівняння першого порядку, що не розв'язні відносно похідної та математичні моделі в яких вони виникають. Математичні моделі фізичної кінетики. Алгоритми дослідження кінетичних систем на основі аналізу графіків реакцій за допомогою ЕОМ.
3. Диференціальні рівняння першого порядку, що не розв'язні відносно похідної та математичні моделі в яких вони виникають. Інтегральні рівняння. Дифро-інтегральні рівняння та методи їх чисельного інтегрування. Диференціальні рівняння вищих порядків, що інтегруються в квадратурах. Поняття про імітаційні математичні моделі. Моделі динаміки росту економіки, популяції їх реалізація на ЕОМ.
4. Лінійні диференціальні рівняння вищих порядків. Типові крайові умови для рівняння теплопровідності. Лінійні крайові задачі та застосування математичних пакетів до опису математичних моделей у яких вони виникають. Загальний розв'язок лінійного однорідного рівняння вищого порядку. Чисельне інтегрування диференціальних рівнянь вищих порядків. Основні поняття теорії різницевого схем. Метод лінійної та градієнтної прогонки.
5. Лінійні однорідні та неоднорідні рівняння зі сталими коефіцієнтами. Моделі коливальних механічних систем де вони виникають. Коливальні моделі для

ізотропного та анізотропного континууму. Використання математичних пакетів для симуляції коливальних процесів. Метод фундаментальних функцій Коші для побудови розв'язку нелінійного однорідного рівняння з сталими коефіцієнтами. Побудова різницевої схеми за допомогою варіаційних принципів.

6. Метод варіації довільної сталої побудови частинного розв'язку лінійного неоднорідного диференціального рівняння. Стаціонарне та нестаціонарне рівняння Шредінгера. Математичні моделі резонансного тунелювання у напівпровідникових наносистемах. Граничні умови, для хвильових функцій.
7. Лінійні диференціальні рівняння n -го порядку. Фундаментальна система розв'язків. Метод варіації довільної сталої. Метод скінченних різниць. Рівномірна та нерівномірна сітки. Лінеаризовані різницеві схеми.
8. Системи лінійних однорідних диференціальних рівнянь зі сталими коефіцієнтами. Матричний метод інтегрування лінійних диференціальних рівнянь зі сталими коефіцієнтами.
9. Різницеві задачі Діріхле для рівняння Пуассона з стаціонарними граничними умовами.
10. Системи лінійних однорідних диференціальних рівнянь зі сталими коефіцієнтами.
11. Моделі масоперенесу у нанопористих структурах. Операторно-різницева схема для рівняння Фіка.
12. Лінійні неоднорідні системи диференціальних рівнянь.
13. Різницева схема Кранка-Ніколсона знаходження самоузгодженого розв'язку системи рівнянь Шредінгера та Пуассона.
14. Побудова загального розв'язку неоднорідної системи рівнянь. Метод фундаментальних функцій Коші.
15. Багатошарові різницеві схеми із знаходження власних значень для змішаних крайових задач.
16. Побудова та алгоритмізація математичних моделей процесів і об'єктів (фільтрації, тепло- та масопереносу, електронного тунельного транспорту та ін.).
17. Рекурсивні алгоритми у математичних пакетах для моделювання розв'язків систем диференціальних рівнянь.
18. Математичні моделі, що зводяться до чисельного розв'язування диференціальних рівнянь першого і другого порядку.

Рекомендована література:

1. Joel S. Cohen. Computer Algebra and Symbolic Computation: Elementary Algorithms. — A K Peters, 2002. P. 1104.
2. Петрик М.Р., Бойко І.В. Диференціальні рівняння у математичних моделях. (Моделювання у середовищі Wolfram Mathematica): навчальний посібник. Тернопіль: : ТНТУ 2016 – 170 с
3. R. Maeder. Computer Science with Mathematica: Theory and Practice for Science, Mathematics, and Engineering. Cambridge University Press, 2000. P.1706.
4. Frank R. Giordano Maurice D. Weir Steven B. Horton William P. Fox. A First Course in Mathematical Modeling, Fourth Edition- Brooks/Cole, 2009. P. 418.
5. Гудименко Ф.С., Павлюк І.А, Волкова В.О. Збірник задач з диференціальних рівнянь. – Київ, Вища школа, 1972. -154 с.

6. Edward B. Magrab. Advanced Engineering Mathematics with Mathematica. CRC Press, 2009. P. 1271.
7. Paul Wellin, Richard Gaylord, Samuel Kamin. An Introduction to Programming with Mathematica, Third Edition. -Cambridge University Press, 2019. P. 804.
8. L. Alves, M. Mikhailov, Renato Cotta. Applied Numerical Analysis with Mathematica. - E-papers Servicos Editoriais, 2017. P. 646.
9. Christian Constanda, Dale Doty, William Hamill. Boundary Integral Equation Methods and Numerical Solutions: Thin Plates on an Elastic Foundation. –Springer, 2016. P. 480.
10. Randall J. SwiftStephen A. Wirkus. A Course in Ordinary Differential Equations, second edition. CRC Press, 2011. P. 504.
11. Cole, J.D., Perturbation Methods in Applied Mathematics, Waltham, Blaisdell Publ. Comp., 1968.
12. Prudnikov, A.P., Brychkov, Yu.A., and Marichev, O.I., Integrals and Series, Vol. 3, More Special Functions, New York, Gordon & Breach Sci. Publ., 1988.
13. Phil Gregory. Bayesian Logical Data Analysis for the Physical Sciences: A Comparative Approach with Mathematica Support. Cambridge University Press, 2010, P. 560.
14. Esa Uusipaikka. Confidence Intervals in Generalized Regression Models. - Chapman & Hall/CRC. 2016. – P. 256.
15. Thomas Colignatus. Conquest of the Plane, Using The Economics Pack Applications of Mathematica for a didactic primer on Analytic Geometry and Calculus. –Thomas Colignatus, 2016. – P. 312.

4. МОДЕЛЮВАННЯ СКЛАДНИХ РОЗПОДІЛЕНИХ СИСТЕМ ТА ОБ'ЄКТІВ

Зміст дисципліни:

Метою дисципліни "Моделювання складних розподілених систем та об'єктів" є ознайомлення студентів з основами вибраного ними професійного напрямку підготовки та надання їм базових знань у студентів із ключових теоретичних і практичних понять та основних перспективних напрямків математичного моделювання та обчислювальних методів. Сформувані основи інструментальних засобів побудови розв'язків математичних моделей прикладного характеру, що описуються крайовими задачами для диференціальних рівнянь у частинних похідних.

Завданням дисципліни є формування прикладного математичного фундаменту бакалавра комп'ютерних технологій, що виражається у спроможності застосовувати і розвивати отримані знання для вирішення проблем наукових досліджень, проектування та експлуатації систем обробки інформації та прийняття рішень, задач, що потребують використання сучасних методів математичного моделювання та інформаційних технологій.

Дисципліна призначена для того, щоб навчити студентів працювати в спеціалізованих програмних системах та системах символічної математики з використанням засобів паралельного програмування та розподілених і кластерних

обчислень. Також безпосередньою метою курсу є ознайомлення студентів з основами наукової роботи за профілем кафедри програмної інженерії.

Тематика завдань:

1. Загальні підходи до класифікації диференціальних рівнянь у частинних похідних і постановка початково-крайових задач.
2. Математичні моделі, що описуються диференціальними рівняннями у частинних похідних.
3. Класифікація рівнянь у частинних похідних. Приведення рівнянь у частинних похідних до канонічного вигляду. Аналіз методів розв'язування рівнянь у частинних похідних та крайових задач для них.
4. Постановка крайової задачі для диференціального рівняння в частинних похідних процесу коливань струни. Постановка початкових і крайових умов для задач і математичних моделей, що описуються рівняннями в частинних похідних.
5. Моделювання і симуляція процесів коливання струни у середовищах математичних пакетів.
6. Метод Даламбера у розв'язуванні крайової задачі для рівняння в частинних похідних, що описує коливання струни та пружної пластинки. Алгоритми реалізації методу у середовищах математичних пакетів.
7. Метод Фур'є для розв'язування рівняння в частинних похідних у задачі про вільні та вимушені коливання. Алгоритми реалізації методу та візуалізація процесів вільних та вимушених коливань.
8. Однорідні лінійні диференціальні рівняння в частинних похідних і властивості їх розв'язків. Процеси поширення тепла в лінійних зразках. Однорідне рівняння Шредінгера та крайові задачі, що виникають у процесах електронного тунелювання.
9. Диференціальні рівняння в частинних похідних у задачах опису процесів вимушених коливань.
10. Розв'язання різних типів диференціальних рівнянь у частинних похідних різними методами. Рівномірна та нерівномірна сітки. Лінеаризовані різницеві схеми.
11. Методи інтегральних перетворень Фур'є для крайових задач диференціальних рівнянь у частинних похідних для обмежених неоднорідних середовищ.
12. Методи інтегральних перетворень Фур'є для крайових задач диференціальних рівнянь у частинних похідних для напівобмежених середовищ.
13. Методи інтегральних перетворень Ганкеля для крайових задач диференціальних рівнянь у частинних похідних для обмежених циліндричних середовищ.
14. Методи інтегральних перетворень Фур'є-Бесселя для крайових задач диференціальних рівнянь у частинних похідних для обмежених циліндричних середовищ.
15. Методика застосування інтегральних перетворень Фур'є-Бесселя для крайових задач моделювання в неоднорідних циліндричних середовищах.
16. Метод інтегрального перетворення Лапласа для розв'язування крайових задач диференціальних рівнянь у частинних похідних для однорідних середовищ.
17. Метод інтегрального перетворення Лапласа для розв'язування крайових задач для диференціальних рівнянь, що виникають у задачах електронного транспорту.

18. Системи диференціальних рівнянь у частинних похідних у математичному моделюванні дифузії газів у пористих середовищах.
19. Системи диференціальних рівнянь у частинних похідних у математичному моделюванні процесів перенесення вологи в середовищі вологомістких частинок.
20. Системи диференціальних рівнянь у частинних похідних у математичному моделюванні фізичних процесів і явищ.
21. Математичне моделювання кінетики масопереносу у пористих середовищах.
22. Математичне моделювання електронних схем нанорозмірних об'єктів методом скінченних різниць.
23. Математичне моделювання складних систем із нелінійними зв'язками.

Рекомендована література:

1. Ляшко І.І., Боярчук О.К., Гай Я.Г., Калайда О.Ф. Диференціальні рівняння. - Київ, Вища школа, 1981. - 504 с.
2. David J. Blower. Information Processing Volume I: Boolean Algebra, Classical Logic, Cellular Automata, and Probability Manipulations. - David J. Blower, Third Millennium Inferencing, 2017. – P. 564.
3. Kevin Hastings. Introduction to Probability with Mathematica, Second Edition Introduction to Probability with Mathematica, Second Edition. - CRC Press, 2012. – P. 291.
4. Самойленко А.Ф., Кривошея С.А., Перестюк М.О., Диференціальні рівняння у прикладах і задачах. - Київ, Вища школа, 2004. - 454 с.
5. Самойленко А.М., Перестюк М.О., Перестюк М.Ю. Диференціальні рівняння. -Київ, Либідь, 2004. - 360 с.
6. Madureira, Alexandre L. Numerical Methods and Analysis of Multiscale Problems. Springer, 2017.- 760 p.
7. Самойленко А.Ф., Кривошея С.А., Перестюк М.О., Диференціальні рівняння у прикладах і задачах. - Київ, Вища школа, 2004. - 454 с.
8. Самойленко А.М., Перестюк М.О., Перестюк М.Ю. Диференціальні рівняння. -Київ, Либідь, 2004. - 360 с.
9. Петрик М., Баб'юк М. Основи математичного моделювання в наукових дослідженнях.- Тернопіль: Підручники і посібники, 1998.- 120 с.
10. Петрик М. MathCAD-технології в інженерних задачах теорії розрахунку і конструювання. - Тернопіль: Вид-во ТДТУ ім. Ів. Пулюя.- 2000 .-154с.
11. Ленюк М.П., Петрик М.Р. Інтегральні перетворення Фур'є, Бесселя із спектральним параметром в задачах математичного моделювання масопереносу в неоднорідних середовищах. Київ: Наукова думка.-2000.-372с.
12. Ляшенко І.М., Мукоєд А.П. Моделювання біологічних та екологічних процесів. – К.: ВПЦ “Київ. ун-т”, 2002.

5. МОДЕЛЮВАННЯ ТА АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Зміст дисципліни:

Метою викладання дисципліни “Моделювання та аналіз програмного забезпечення” є розвинення у студентів навичок моделювання програмного забезпечення шляхом аналізу предметної області, застосування методів декомпозиції складної системи та аналізу взаємозв’язків між її елементами, абстрагування та класифікація сутностей предметної області для встановлення ієрархії класів, що реалізують функціональності системи для конкретних представників (екземплярів) користувача, використовуючи сучасні методи та інструментальні засоби моделювання.

Завданням вивчення дисципліни є набуття студентами практичних навичок та теоретичних знань з моделювання та аналізу програмного забезпечення з використанням засобів UML.

Тематика завдань:

1. Основні підходи та концепції моделювання ПЗ. Основи проєктування ПЗ. Аналіз Вимог.
2. Основні етапи розробки ПЗ. Життєвий цикл ПЗ.
3. Формалізовані мови моделювання та сучасні засоби проєктування ПЗ.
4. Інструментальні засоби розробки ПЗ. Нові підходи і принципи проєктування ПЗ: "Від вимог до коду".
5. Моделі вимог. Методологія моделювання варіантів використання
6. Ідентифікація класів предметної області на основі Специфікації вимог та описів ВВ. Моделювання класів на UML
7. Моделювання в UML відношень та зв’язків між класифікаторами (класами, ВВ, підсистемами, компонентами).
8. Архітектурний підхід до проєктування діаграм ієрархії класів.
9. Детальне проєктування класів. Розробка деталізованого опису класу на UML та IBM RSA. Моделювання атрибутів та операцій класів
10. Моделювання взаємодій системи з користувачем через використання діаграм взаємодій.
11. Діаграми послідовностей (Sequence Diagram SD). Моделювання повідомлень як взаємодії об’єктів. Діаграми комунікацій (Communication Diagram)
12. Моделювання сценаріїв реалізації ВВ на основі UML та IBM RSA. Концепція кооперації об’єктів класів, що обмінюються повідомленнями. Створення діаграм послідовностей SD- та діаграм кооперації CD-діаграм для реалізації сценаріїв ВВ засобами IBM RSA у максимально формалізованій алгоритмічній формі для їх реалізації в програмний код.

13. Моделювання потоків керування. Діаграми діяльності.
14. Моделювання об'єктів. Класи і відношення між ними. Структура та проектування складних систем
15. Діаграми станів. Скінченні автомати. Моделювання поведінки спільно працюючих об'єктів у сценаріях варіантів використання.
16. Моделювання інтерфейсів для побудови компонентно-орієнтованої архітектури ПЗ.
17. Моделювання кооперацій, компонентів та розміщення вузлів обладнання.
18. Моделі та підходи проектування ПЗ, що ґрунтуються на використанні узагальнених проектних рішень. Узагальнене проектування та вимоги у вигляді класів проектування, кооперацій, підсистем. Декомпозиція системи на підсистеми.
19. Вибір концептуальної моделі розробки ПЗ на основі аналізу вимог предметної області (згідно варіантів)
20. Аналіз сутностей предметної області як кандидатів в класи системи.
21. Практичні приклади моделей предметної області (МПО). Реалізація МПО через ідентифікацію класів, що реалізують усі необхідні для користувача функціональні операції. Приклад моделі глосарію предметної області. Визначення зв'язків між структурними сутностями.
22. Методи визначення кандидатів в класи, методи і атрибути. Розширене моделювання відношень між класами (способи взаємодії і комбінування абстракцій): на базі використання засобів декомпозиції та ОО-аналізу - успадкування, залежності, асоціації (агрегації).
23. Поглиблене розуміння суті об'єктів та їх станів в моделюванні поведінки ПЗ. Створення і ліквідація об'єктів на ЛЖ. Суть повідомлень різних типів (завершені, загублені, знайдені, повідомлення повернення) як способів взаємодії об'єктів і ролей.
24. Розширене розуміння діаграм послідовностей щодо практичного використання. Різні типи взаємодій об'єктів і прототипних ролей на. Фокус керування. Основний зміст діаграми послідовності і комунікації. Їх принципові відмінності. Впорядковані (за часом та об'єктами) взаємодії на ЛЖ. Види керування на діаграмах послідовності.
25. Практичні аспекти проектування інтерфейсів (І). І, реалізований компонентом або класом (надаваний). Необхідний інтерфейс компоненти або класу. Розширені зв'язки між компонентом та її інтерфейсами. Замінність. Приклади компонентів і розміщення для різних предметних областей.
26. Механізми UML-пакування як засоби вирішення проблеми складності системи. Пакети як підсистеми, що містять компоненти ПЗ. Суть і необхідність компонентної розробки ПЗ. Внутрішня структура пакетів, підсистем компонент.

27. Відповідальність класів . Вимоги проектування класів як елементів архітектури та реалізаторів варіантів використання системи. Розширене розуміння кооперації класів. Класи аналізу і проектування.
28. Засоби забезпечення прийняття стандартизованих рішень при проектуванні ПЗ. Застосування зразків і шаблонів проектування різних категорій в створенні моделей предметних областей. Переваги та недоліки патернів різних категорій (патерни створення та поведінки. Приклади їх структури. Імплементация в Java- або C++-код.
29. Системна архітектура програмної системи. Ознаки складної системи
30. Декомпозиція систем. Структура системи. Категорії класів та складові частини об'єктно-орієнтованого підходу.
31. Специфікації. Об'єктна модель
32. UML як мова моделювання ПЗ. Структура системи. Складові частини об'єктно-орієнтованого підходу Об'єктна модель. Складові частини об'єктно-орієнтованого підходу

Рекомендована література:

1. Rumbaugh J., Jacobson I., Booch G. UML 2.0, guide de référence. Editions Campus Press (2005).
2. Benoît Charoux, Aomar Osmani, Yann Thierry-Mieg. UML 2.0. Editions Pearson, Education France (2008).
3. UML 2.0 Superstructure et_ UML 2.0 Infrastructure OMG (Object Management Group). www.uml.org (2004).
4. Gamma, E., Helm, R., Johnson, R., and Vlissides, J. 1995. Design Patterns: Elements of Reusable Object-Oriented Software. Reading, MA: Addison-Wesley.
5. Hensgen P. Підручник з Umbrello UML Modeller [Електронний ресурс] / P. Hensgen, Ю. Чорноіван. – 2013. – Режим доступу до ресурсу: <https://docs.kde.org/stable5/uk/kdesdk/umbrello/index.html>
6. Grady Booch, Ivar Jacobson, and James Rumbaugh. Object-Oriented Analysis and Design with Applications [Електронний ресурс] / Grady Booch, Ivar Jacobson, and James Rumbaugh – Режим доступу до ресурсу: <https://zjnu2017.github.io/OOAD/reading/Object.Oriented.Analysis.and.Design.with.Applications.3rd.Edition.by.Booch.pdf>.
7. Специфікації UML. [ABOUT THE UNIFIED MODELING LANGUAGE SPECIFICATION VERSION 2.5.1 Електронний ресурс] / – Режим доступу до ресурсу: <https://www.omg.org/spec/UML/2.5.1/PDF>.
8. Г.В. Табунщик. Проектування та моделювання програмного забезпечення сучасних інформаційних систем [Електронний ресурс] / Г.В. Табунщик, Т.І. Каплієнко, О.А. Петрова. – 2016. – Режим доступу до ресурсу:

- http://eir.zntu.edu.ua/bitstream/123456789/1824/1/Tabunshchik_Software_Design.pdf 2.
9. Практикум з об'єктно-орієнтованих методологій створення комп'ютерних систем: Методичні вказівки до виконання практичних робіт для студентів напряму «Комп'ютерні науки» // Укладач В.О. Дегтярьов. – Миколаїв: Видавництво ЧДУ ім. Петра Могили, 2009. – Вип. 134. – 68 с. (Методична серія)
 10. Object-Oriented Analysis, Design and Implementation An Integrated Approach [Електронний ресурс] / Brahma Dathan, Sarnath Ramnath – Режим доступу до ресурсу:
https://warin.ca/ressources/books/2015_Book_ObjectOrientedAnalysisDesignA.pdf.
 11. Петрик М., Мудрик І., Петрик О., Ю. Стоянов. Сучасні технології ООП-проектуювання та автоматичного генерування програмного коду: навчальний посібник, Тернопіль: ТНТУ імені Івана Пулюя, 2018. 48 с.
 12. Моделювання та аналіз програмного забезпечення ID: 1351 2. Петрик М. Моделювання та аналіз програмного забезпечення. Науково-методичний посібник. / Петрик М., Петрик О. - Тернопіль: Вид-во ТНТУ ім. І. Пулюя.-2014.-180с.
 13. Петрик М.Р. Моделювання програмного забезпечення: науково-методичний посібник / М.Р. Петрик, О.Ю. Петрик – Тернопіль : Вид-во ТНТУ імені Івана Пулюя, 2015. – 200 с. URL: <http://elartu.tntu.edu.ua/handle/123456789/17796>
 14. Rational Software Architect Designer 9.7.0 – Режим доступу до ресурсу: <https://www.ibm.com/support/pages/rational-software-architect-designer-970>

6. ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ ТА ОБ'ЄКТНЕ МОДЕЛЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ НА ОСНОВІ OCL

Зміст дисципліни:

Метою викладання дисципліни “Об'єктно-орієнтоване програмування та об'єктне моделювання програмного забезпечення на основі OCL” є вивчення сучасного підходу до розробки програмного забезпечення на основі об'єктно-орієнтованих парадигм і технологій, набуття студентами знань про об'єктно-орієнтований підхід у програмуванні, засвоєння концепції об'єктно-орієнтованого підходу та його практичне застосування, освоєння можливостей мови C++ з акцентуванням уваги на вирішенні об'єктно-орієнтованих проблем проектування сучасних програмних систем. Вивчення елементів теорії і практики моделювання програмного забезпечення з використанням UML, розвинення у студентів систематизованого представлення і практичних навичок застосування мови моделювання UML і відповідного об'єктно-орієнтованого підходу до моделювання програмного забезпечення.

Тематика завдань:

1. Поняття об'єктно-орієнтованого програмування. Визначення програмування на основі абстрактних типів даних Інструментальні засоби ООП: інкапсуляція, поліморфізм, наслідування.
2. Загальний формат оголошення класу. Загальний формат визначення методів класу. Різниця між класом, оголошеним зі службовим словом `struct` і класом, оголошеним зі службовим словом `class`
3. Що таке об'єкт? Як оголосити об'єкт класу? Яка умова є обов'язковою для присвоєння одного об'єкта іншому?
4. Чи можна адресу об'єкта передати у функцію в якості аргументу?
5. Що резервує пам'ять оголошення чи визначення класу?
6. Яка область дії членів класу, визначених з атрибутом `public`?
7. Що означає атрибут `private` при оголошенні класу?
8. Хто має доступ до даних класу, оголошених з атрибутом `protected`?
9. Дайте визначення конструктора. Коли він викликається?
10. Що таке деструктор? Коли він викликається?
11. Скільки конструкторів можна оголосити в класі?
12. Скільки деструкторів можна оголосити в класі?
13. Які Ви знаєте види конструкторів?
14. Які Ви знаєте властивості конструкторів?
15. Що таке конструктор копіювання?
16. У яких випадках викликається конструктор копіювання?
17. Що таке конструктор з параметрами? Що таке конструктор по замовчуванню? Сформулюйте основні властивості конструкторів.
18. Який тип можуть мати параметри конструктора?
19. Що таке вказівник `this`? Яка його функція при ООП?
20. Чи правильне твердження, що для кожного об'єкта класу створюється своя копія даних класу, а методи класу існують для всіх об'єктів у одному екземплярі?
21. Наведіть приклад використання вказівника `this`.
22. Як задати параметри для методу по замовчуванню?
23. Дайте визначення дружньої функції.
24. Як оголосити дружню функцію?
25. Дружня функція є методом класу?
26. Чи може дружня функція бути дружньою більше, ніж до одного класу?

27. Чи істинне твердження, що дружня функція має доступ до закритих даних класу, навіть не будучи його методом?
28. Що таке дружній клас?
29. Які Ви знаєте способи перевантаження функцій?
30. Запишіть основну форму унарної оператор-функції методу класу
31. Запишіть основну форму унарної оператор-функції дружньої до класу
32. Запишіть загальний формат бінарної оператор-функції методу класу
33. Запишіть загальний формат бінарної оператор-функції дружньої до класу
34. Які Ви знаєте обмеження на перевантаження операторів?
35. Що таке базовий клас?
36. Що таке похідний клас?
37. Що таке безпосередній базовий клас?
38. Які Ви вивчали приклади поліморфізму?
39. Запишіть формат опису конструктора при множинному наслідуванні
40. Поясніть різницю між простим і множинним наслідуванням.
41. Який атрибут має перевагу: атрибут при наслідуванні чи атрибут у безпосередньому базову класі?
42. Який формат опису класу при простому наслідуванні?
43. Який клас називається абстрактним?
44. Який клас називається поліморфним?
45. У базовому класі встановлений статус доступу `public`. Яким має бути модифікатор доступу, щоб в похідному класі одержати статус доступу `protected`?
46. Чи є кожен об'єкт похідного класу об'єктом відповідного базового класу?
47. Чи є кожен об'єкт базового класу об'єктом класів, породжених даним базовим класом?
48. Який формат опису класу при множинному наслідуванні?
49. Поясніть різницю між дружньою, віртуальною і перевантаженою функціями
50. Які Ви знаєте атрибути доступу при наслідуванні?
51. Як відбувається виклик конструкторів при наслідуванні?
52. Як відбувається виклик деструкторів при наслідуванні?
53. Запишіть формат опису конструктора при простому наслідуванні
54. Наведіть приклад ситуації, що приводить до необхідності оголошення класу віртуальним.

55. Які Ви знаєте види множинного наслідування? Охарактеризуйте їх (можна схематично).
56. Нехай базовий клас містить метод `func()`, а похідний клас не має методу з таким ім'ям. Чи може об'єкт похідного класу мати доступ до методу `func()`?
57. Допустимо, що базовий і похідний класи включають методи з однаковими іменами. Який з методів буде викликаний об'єктом похідного класу, якщо не використана операція дозволу імені?
58. Наведіть приклад ситуації, що приводить до необхідності використання віртуальної функції.
59. Визначення потоку. Класифікація потоків введення / виведення за напрямком передачі даних і за способом створення
60. Схема наслідування класів потоків
61. Засоби форматування потоків виведення
62. Загальний формат і призначення функції `width()`. Навести приклад використання
63. Загальний формат і призначення функції `precision()`. Навести приклад використання
64. Загальний формат і призначення функції `fill()`. Навести приклад використання
65. Загальний формат і призначення функції `setf()`. Навести приклад використання
66. Загальний формат і призначення функції `unsetf()`. Навести приклад використання
67. Форматування потоків введення / виведення за допомогою маніпуляторів
68. Маніпулятори введення, визначені користувачем. Загальний формат
69. Маніпулятори виведення, визначені користувачем. Загальний формат
70. Схема роботи з файлами через потоки
71. Створення потоку введення, виведення, введення-виведення.
72. Функції, що дозволяють зв'язати потік з файлом. Прототип функції `open()`. Режими відкриття файлу через потоки. Методи для запису у файл `write` і `put`: прототип, приклад.
73. Методи переміщення поточної позиції введення (виведення) `seekp` і `seekg`: прототип, приклад
74. Методи визначення позиції покажчика зчитування чи запису `tellg()` і `tellp()`. Методи читання з файлу `read()` і `get()`: прототип, приклад. Які Ви знаєте способи форматування потоків?
75. Шаблон функції. Загальна форма визначення шаблону функції
76. Кількість та типи параметрів шаблону функції
77. Шаблон простої функції. Шаблони функцій з параметрами. Вимоги до фактичних параметрів шаблону

78. Основні властивості параметрів шаблону функції. Способи перевантаження шаблонних функцій. Загальна форма визначення шаблону класу. Оголошення об'єкта шаблону класу. Шаблони і наслідування. Шаблони і друзі.
79. Поняття механізму виключних ситуацій. Приклади виключень.
80. Випадки, коли необхідно передбачити опрацювання виключних ситуацій
81. Оператори для реалізації механізму обробки виключень. Оператор перехвату виключень `try`. Оператори обробки виключень `catch`. Оператори викиду виключень `throw`. Синтаксис і семантика генерації обробки виключень
82. Введення та виведення в Qt у візуальному режимі.
83. Поняття про класи та об'єкти. Базові принципи ООП.
84. Оголошення класів та об'єктів.
85. Робота з графікою в Qt, координати у графічному режимі. Двовимірна графіка.
86. Відмінності C++ від C.
87. Логічний тип даних у C++. Старий та новий стиль C++. Новий стиль заголовків C++ та простір імен.
88. Оператори введення-виведення `<<`, `>>`.
89. Вбудовані потоки у C++. Форматоване введення-виведення C++.
90. Форматування за допомогою членів класу `ios`. Прапорці форматування.
91. Встановлення та скидання прапорців форматування. Ширина поля введення, точність та заповнення.
92. Використання маніпуляторів формату. Маніпулятори формату.
93. Робота з масивом у Qt.
94. Відмінність підпрограм у C та C++. Підпрограми в Qt з використанням ООП.
95. Рядки у C++. Об'єкти класу `string`. Задавання рядків. Введення та виведення рядків.
96. Рядки у C++. Об'єкти класу `string`. Операції для роботи з рядками.
97. Рядки у C++. Методи для роботи з рядками. Присвоєння, додавання, порівняння та перетворення рядків.
98. Рядки у C++. Методи для роботи з рядками. Пошук у рядку. Отримання характеристик рядків.
99. Робота з файлами у C++. Потоки для роботи з файлами. Створення файлових потоків.
100. Робота з файлами у C++. Режим відкривання файлів. Відкривання та закривання файлів.
101. Робота з файлами у C++. Читання та запис файлів. Методи читання потоків.
102. Робота з файлами у C++. Читання та запис файлів. Методи запису потоків.

103. Робота у візуальному режимі Qt. Поняття про сигнали та слоти.
104. Робота у візуальному режимі Qt. Підключення сигналів та слотів.
105. Робота у візуальному режимі Qt. Створення своїх сигналів.
106. Робота у візуальному режимі Qt. Створення своїх слотів.
107. Робота у візуальному режимі Qt. Поняття про події та взаємодія з користувачем.
108. Робота у візуальному режимі Qt. Події введення.
109. Робота у візуальному режимі Qt. Перезавантаження методів обробки подій. Події клавіатури.
110. Робота у візуальному режимі Qt. Клас QFocusEvent.
111. Робота у візуальному режимі Qt. Подія оновлення контексту малювання. Події. Події таймера. Таймер.
112. Робота у візуальному режимі Qt. Збереження працездатності програми в критичних режимах. Додавання іконок до вікна та файлу, що виконується.

Рекомендована література:

1. Williams A., C++ Concurrency in Action - Manning Publications Co., 2019. – 592p.
2. Dmitrović S., Modern C++ for Absolute Beginners: A Friendly Introduction to C++ Programming Language and C++11 to C++20 Standards – Apress, 2020. – 461p.
3. Horton's I. Beginning Visual C++® 2013 [Електронний ресурс] / Ivor Horton's – Режим доступу до ресурсу: <https://www.pdfdrive.com/ivor-hortons-beginning-visual-c-2013-d46824353.html>.
4. Horton's I. Beginning C++17 From Novice to Professional / [Електронний ресурс] / Ivor Horton's – Режим доступу до ресурсу: <https://www.pdfdrive.com/beginning-c17-from-novice-to-professional-e184792973.html>.
5. Stroustrup B. Programming: Principles and Practice Using C++ [Електронний ресурс] / Stroustrup Bjarne. – 2014. – Режим доступу до ресурсу: <https://www.pdfdrive.com/programming-principles-and-practice-using-c-e18903967.html>.
6. R. Lafore. Object-oriented programming in C ++. / [Електронний ресурс] / R. Lafore. – Режим доступу до ресурсу: <https://docs.google.com/file/d/0B21HoBq6u9TsUHhqS3JIUmFuamc/view?sourcekey=0-MYlet9RIjEukd6CvLEHUbW>
7. Об'єктно-орієнтоване програмування: [Підручник] / В.В. Бублик. – К.: ІТ- книга, 2015. – 624 с.: іл. [Електронний ресурс]/ В.В. Бублик. – Режим доступу до ресурсу: <https://docs.google.com/file/d/0B21HoBq6u9TsUHhqS3JIUmFuamc/view?sourcekey=0-MYlet9RIjEukd6CvLEHUbW>

8. О. Петрик. Об'єктно-орієнтоване програмування в середовищі C++: Навчальний посібник. Лабораторний практикум Тернопіль, видавництво ТНТУ імені Івана Пулюя, 2011. 188 с.
9. Browning B., C++20 Recipes: A Problem-Solution Approach - Apress; 2nd ed. edition, 2020 -657p.
10. Gregoire M., Professional C++, 5th Edition - Wrox; 5th edition, 2021 – 1312p.
11. Lospinoso J., C++ Crash Course: A Fast-Paced Introduction - No Starch Press; Illustrated edition 2019 – 792p.
12. Bancila M., Modern C++ Programming Cookbook - Packt Publishing; 2nd ed. edition, 2020 – 750p.
13. Solter N. A., Kleper S. J., Professional C++: Second Edition, 2022 – 1104p.
14. Swaminathan J., Mastering C++ Programming: Modern C++17/20 Features - Packt Publishing, 2021 – 559p.
15. Sehr V., C++20 Revealed: Discover the Most Modern, Effective, and Safe Programming Language – Apress, 2020 – 547p.
16. freeCodeCamp.org Object Oriented Programming (OOP) in C++ Course [Електронний ресурс] // freeCodeCamp.org – Режим доступу до ресурсу: <https://www.youtube.com/watch?v=wN0x9eZLix4>.
17. Darshan University. Concepts of OOP [Електронний ресурс] // Darshan University. – Режим доступу до ресурсу: <https://www.youtube.com/watch?v=hyJeuFWmROc&list=PLftJ4X48yC1k97-tmIpp0HpW-lh2dv9i2>.
18. STL <http://community.topcoder.com/tc?module=Static&d1=tutorials&d2=standardTemplateLibrary>
19. Стандартна бібліотека. Довідник <http://www.cplusplus.com/reference/>
20. Введення в об'єктно-орієнтоване програмування <http://cppstudio.com/uk/cat/274/281/>
21. Довідник по C++ [Електронний ресурс]. - Режим доступу: URL: <https://msdn.microsoft.com/uk-ua/library/3bstk3k5.aspx>
22. Система онлайн-тестування [Електронний ресурс]. - Режим доступу: URL: <http://www.quizful.net/category/cpp> 17
23. Сайт з найновішою довідкою по C++ [Електронний ресурс]. - Режим доступу: URL: <http://www.cplusplus.com>
24. Visual Studio Team Services [Електронний ресурс]. - Режим доступу: URL: <https://learn.microsoft.com/uk-ua/>
25. IDE Visual C++ [Електронний ресурс]. - Режим доступу: URL: <https://support.microsoft.com/uk-ua/help/2977003/the-latest-supported-visualc-downloads>

V. ПОРЯДОК ПРОВЕДЕННЯ ТА КРИТЕРІЇ ОЦІНЮВАННЯ ВСТУПНОГО ІСПИТУ

Форма проведення вступного іспиту зі спеціальності – письмова у вигляді тестових завдань.

Структура тестових завдань передбачає 6 тематичних блоків із вибором однієї правильної відповіді у кожному тестовому завданні (30 тестових завдань). Кожне тестове завдання оцінюється в один бал. Час на виконання - 120 хвилин.

За результатами вступного іспиту вступник отримує від 0 до 30 тестових балів, котрі переводяться в рейтингову оцінку від 100 до 200 балів відповідно до таблиці:

Кількість тестових балів	Рейтингова оцінка	Кількість тестових балів	Рейтингова оцінка
0-9	не склав	20	150
10	100	21	155
11	105	22	160
12	110	23	165
13	115	24	170
14	120	25	175
15	125	26	180
16	130	27	185
17	135	28	190
18	140	29	195
19	145	30	200

VI. ВИМОГИ ДО НАПИСАННЯ ДОСЛІДНИЦЬКОЇ ПРОПОЗИЦІЇ

Дослідницька пропозиція є самостійною авторською роботою, у якій обґрунтовано мотив вибору майбутнього дисертаційного дослідження з метою розв'язання актуальної наукової проблеми.

За обсягом дослідницька пропозиція має становити 5-10 сторінок машинописного тексту формату А4 (**основна частина**), розмір шрифту 14, інтервал – 1,5, абзац – відступ 1 см, поля: верх, низ – 2 см, зліва 3 см, праворуч 1 см, нумерація сторінок – зверху праворуч.

Подана дослідницька пропозиція не обов'язково саме у такому формулюванні стане темою дисертаційного дослідження вступника у випадку його зарахування до аспірантури.

Дослідницька пропозиція обов'язково має містити:

- обґрунтування актуальності та практичної важливості досліджуваної проблеми/завдання;
- окреслення потенційних дослідницьких питань, відповіді на які вступник прагне віднайти;

- формулювання мети і завдань, об'єкта, предмета і методів дослідження, опис інформаційної бази дослідження;

- висновки щодо майбутнього наукового дослідження шляхом прогнозування наукових результатів, що можуть бути досягнуті.

Структурними елементами дослідницької пропозиції є:

- титульна сторінка (додаток А);
- зміст;
- вступ (актуальність, мета, завдання, об'єкт, предмет, методи дослідження);
- основна частина;
- висновок;
- список використаних джерел (оформити згідно ДСТУ 8302:2015 «Інформація та документація. Бібліографічне посилання. Загальні положення та правила складання»).

Список використаних джерел має свідчити про ознайомлення автора як з фундаментальними науковими працями, так і з останніми публікаціями за обраною темою (вітчизняними і зарубіжними, крім тих, які опубліковані у виданнях держави, визнаної Верховною Радою України державою-агресором).

Дослідницька пропозиція не повинна містити академічного плагіату, фальсифікації, фабрикації та відповідати вимогам Кодексу корпоративної етики Тернопільського національного технічного університету імені Івана Пулюя <https://docs.tntu.edu.ua/base/document?id=461>.

Дослідницька пропозиція подається у паперовому варіанті (за підписом абітурієнта) та електронному (надсилається на електронну скриньку vid_asp@tntu.edu.ua) до відбіркової комісії відділу аспірантури та докторантури разом із іншими документами вступника. Уповноважений працівник відбіркової комісії забезпечує перевірку дослідницької пропозиції щодо оригінальності тексту відповідно до Положення про недопущення академічного плагіату в Тернопільському національному технічному університеті імені Івана Пулюя (https://phd.tntu.edu.ua/wp-content/uploads/2022/12/563930_225275_pologhennya_pro_plagiat.pdf).

При встановленні фактів плагіату дослідницька пропозиція не розглядається, а автор до наступних вступних випробувань не допускається.

Через відповіді абітурієнта на питання стосовно його/її дослідницької пропозиції та обізнаності в обраному науковому напрямку оцінюється базовий рівень готовності вступника до навчання на освітньо-науковій програмі та до продуктивного виконання дисертаційного дослідження.

Автор дослідницької пропозиції має продемонструвати:

- чітке розуміння досліджуваної проблеми/завдання, знання дискусійних питань, пов'язаних з нею/ним;
- володіння методикою проведення науково-дослідницької роботи.
- вміння підбирати, систематизувати та аналізувати фактичний матеріал;
- здатність формулювати обґрунтовані висновки та окреслювати перспективи подальших досліджень;
- уміння мислити раціонально і творчо.

Дослідницька пропозиція може бути запропонована стейкхолдерами, з якими співпрацює вступник.

VII. ОСКАРЖЕННЯ РЕЗУЛЬТАТІВ ВСТУПНИХ ВИПРОБУВАНЬ

Вступник має право на подання апеляції на результат вступного іспиту зі спеціальності чи співбесіди, яка розглядається апеляційною комісією, склад та порядок роботи якої затверджуються наказом ректора. Апеляції подаються вступником особисто в письмовій формі не пізніше наступного дня після оголошення результатів вступного іспиту зі спеціальності чи співбесіди.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Тернопільський національний технічний університет імені Івана Пулюя

ДОСЛІДНИЦЬКА ПРОПОЗИЦІЯ

ТЕМА ДОСЛІДЖЕННЯ

Спеціальність F2 «Інженерія програмного забезпечення»

_____ /ППП/

Тернопіль – 20__ р

ЗАТВЕРДЖЕНО
Наказ Міністерства освіти
і науки України
05.07.2016 № 782

Форма № Н-1.09

ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
ІМЕНІ ІВАНА ПУЛЮЯ

ПРОТОКОЛ СПІВБЕСІДИ № ____

(прізвище, ім'я, по батькові вступника)

пройшов (пройшла) співбесіду « ____ » _____ 20__ року
у відбірковій комісії _____
(найменування (назва) інституту/факультету/відділення)

За підсумками співбесіди комісія вирішила _____ до зарахування
(рекомендувати, не рекомендувати)

аспірантом (аспіранткою) _____ курсу
третього рівня вищої освіти/освітньо-наукового рівня доктор філософії

за спеціальністю _____
(код і назва спеціальності)

Мотивований висновок:

Голова комісії _____
(підпис) _____ (прізвище та ініціали)

Члени комісії: _____
(підпис) _____ (прізвище та ініціали)

(підпис) _____ (прізвище та ініціали)

(підпис) _____ (прізвище та ініціали)

(підпис) _____ (прізвище та ініціали)