МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

ТЕРНОПІЛЬСЬКИЙ НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
імені ІВАНА ПУЛЮЯ

ЗАТВЕРДЖЕНО
Голова приймальної комісії
_____Микола МИТНИК
«02» 06 2026 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 – 2026

ЗМІСТ

І. 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 в аспірантурі та докторантурі Тернопільського національного технічного університету імені Івана Пулюя https://phd.tntu.edu.ua/wp-content/uploads/2025/10/865503_pologhennya_doktor_filosofiyi_092025.pdf та Правил прийому на навчання до Тернопільського національного технічного університету імені Івана Пулюя в 2026 році https://tntu.edu.ua/storage/pages/00000314/tntu-pp2026.pdf.

Вступні випробування для прийому на навчання для здобуття ступеня доктора філософії проводяться як:

- зовнішнє оцінювання у формі єдиного вступного іспиту (ЄВІ), єдиного вступного випробування з методології наукових досліджень (ЄВВ) (див. Український центр оцінювання якості освіти: https://testportal.gov.ua/vstupni-do-aspirantury-2/);
- оцінювання в Університеті у формі іспиту з спеціальності.

ІІ. ЗМІСТ ПРОГРАМИ ІСПИТУ ЗІ СПЕЦІАЛЬНОСТІ

Метою складання вступного іспиту зі спеціальності Інженерія програмного забезпечення є перевірка й оцінювання теоретико-методологічної підготовки вступників, їхніх знань фундаментальних концепцій проєктування складних програмних систем, формальних методів верифікації та математичного моделювання ПЗ, а також здатності аналізувати й оптимізувати процеси життєвого циклу програмних продуктів для вирішення науково-дослідних завдань.

Зокрема, перевірці підлягають знання методологічних засад архітектурного проектування складних розподілених систем, сучасних парадигм програмування, інструментальних засобів автоматизації розробки, а також методів забезпечення якості, надійності та безпеки програмних систем.

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 типів даних; реалізовувати структури даних типу дерево, множина, орієнтований граф, неорієнтований граф. реалізовувати основі 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. О-символіка..

7. Основні правила аналізу часу виконання алгоритмів.

8. Аналіз об’єму пам’яті, що використовуватиметься програмою.

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 відшукання 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 невпорядкованих 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. Aho A. V. Data Structures and Algorithms / A. V. Aho, J. E. Hopcroft, J. D. Ullman. Boston : Pearson, 2010. 448 p.
2. Cormen T. H. Introduction to Algorithms / T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. 3rd ed. Cambridge: MIT Press, 2009. 1312 p.
3. Knuth D. E. The Art of Computer Programming, Volumes 1–4A / D. E. Knuth. Boston: Addison-Wesley Professional, 2011. 3168 p.
4. Sedgewick R. Algorithms / R. Sedgewick, K. Wayne. 4th ed. Upper Saddle River: Addison-Wesley Professional, 2011. 976 p.
5. Sedgewick R. Algorithms in C++, Parts 1–4 : Fundamentals, Data Structure, Sorting, Searching / R. Sedgewick. 3rd ed. Boston : Addison-Wesley Professional, 1998. 738 p.

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 управлінню 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, ISO/IEC). Спеціальні аспекти проєктування, їх деталізація.
7. Опис контексту проєктування: від вимог до тестів.
8. Двокроковий процес проектування. Архітектурне проєктування - декомпозиція структури (*статична*) і організації (*динамічна*) компоненти.
9. Деталізація архітектури (описує специфічну поведінку і характеристики окремих компонентів.
10. Результати процесу проєктування: 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 підходах 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 деталей 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 primitiviness).
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 єдину систему; 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 наступили.
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 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. Діграма 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 актанти
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. Clements P. Documenting Software Architectures : Views and Beyond / P. Clements, F. Bachmann, L. Bass та ін. 2nd ed. Boston : Addison-Wesley Professional, 2010. 592 p.
2. Будай А. Дизайн-патерни — просто, як двері / А. Будай. [Б. м.] : [б. в.], 2012. 90 с.
3. Arlow J. UML 2 and the Unified Process : Practical Object-Oriented Analysis and Design / J. Arlow, I. Neustadt. 2nd ed. Boston : Addison-Wesley Professional, 2005. 624 p.
4. Rumbaugh J. The Unified Modeling Language Reference Manual / J. Rumbaugh, I. Jacobson, G. Booch. 2nd ed. Boston : Addison-Wesley Professional, 2004. 752 p.
5. Booch G. Object-Oriented Analysis and Design with Applications / G. Booch, R. A. Maksimchuk, M. W. Engle та ін. 3rd ed. Boston : Addison-Wesley Professional, 2007. 720 p.
6. UML 2.0 Object Constraint Language (OCL) // Object Management Group (OMG). 2004. URL: http://www.omg.org.
7. Favre J.-M. Cours de Jean-Marie Favre / J.-M. Favre. Paris : IMAG, 2014.
8. Петрик М. Моделювання та аналіз програмного забезпечення / М. Петрик, О. Петрик. Тернопіль : Вид-во ТНТУ, 2015. 210 с.
9. Charoux B. UML 2.0 et UML 2.0 Infrastructure / B. Charoux, A. Osmani, Y. Thierry-Mieg. Paris : Pearson Education France, 2008. 294 p.
10. Alexander I. F. Writing Better Requirements / I. F. Alexander, R. Stevens. London : Addison-Wesley, 2002. 168 p.
11. Ambler S. W. The Unified Process Inception Phase / S. W. Ambler, L. L. Constantine. Lawrence : CMP Books, 2000. 321 p.
12. Ambler S. W. The Unified Process Elaboration Phase / S. W. Ambler, L. L. Constantine, R. Smith. Lawrence : CMP Books, 2000. 400 p.
13. Ambler S. W. The Unified Process Construction Phase / S. W. Ambler, L. L. Constantine. Lawrence : CMP Books, 2000. 400 p.
14. Arlow J. Enterprise Patterns and MDA : Building Better Software with Archetype Patterns and UML / J. Arlow, I. Neustadt. Boston : Addison-Wesley Professional, 2004. 512 p.

15. Frankel D. S. Model Driven Architecture : Applying MDA to Enterprise Computing / D. S. Frankel. Indianapolis : John Wiley & Sons, 2003. 352 p.
16. Kay M. XSLT Programmer's Reference / M. Kay. 2nd ed. Birmingham : Wrox Press, 2001. 972 p.
17. Kleppe A. MDA Explained : The Model Driven Architecture: Practice and Promise / A. Kleppe, J. Warmer, W. Bast. Boston : Addison-Wesley Professional, 2003. 192 p.
18. Kroll P. The Rational Unified Process Made Easy : A Practitioner's Guide to the RUP / P. Kroll, P. Kruchten. Boston : Addison-Wesley Professional, 2003. 416 p.
19. Kruchten P. The Rational Unified Process : An Introduction / P. Kruchten. 2nd ed. Boston : Addison-Wesley Professional, 2000. 320 p.
20. Kruchten P. The 4+1 View of Architecture / P. Kruchten // IEEE Software. 1995. Vol. 12, no. 6. P. 42–50.
21. Leffingwell D. Managing Software Requirements : A Use Case Approach / D. Leffingwell, D. Widrig. 2nd ed. Boston : Addison

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 напряму 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. Сформувати 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 диференційних 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 комп'ютеризованих систем.
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 комп'ютерних системах.
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 різницевих 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 однорідні рівняння з сталими коефіцієнтами. Побудова різницевих схем за допомогою варіаційних принципів.
6. Метод варіації довільної сталої побудови частинного розв'язку лінійного неоднорідного диференціального рівняння. 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. Лінійні неоднорідні системи диференціальних рівнянь.
12. Різницева схема Кранка-Ніколсона знаходження самоузгодженого розв'язку системи рівнянь Шредінгера та Пуассона.
13. Побудова загального розв'язку неоднорідної системи рівнянь. Метод фундаментальних функцій Коші.
14. Багатошарові різницеві схеми із знаходження власних значень для змішаних крайових задач.
15. Побудова та алгоритмізація математичних моделей процесів і об’єктів (фільтрації, тепло- та масопереносу, електронного тунельного транспорту та ін.).
16. Рекурсивні алгоритми у математичних пакетах для моделювання розв'язків систем диференційних рівнянь.
17. Математичні моделі, що зводяться до чисельного розв'язування диференціальних рівнянь першого і другого порядку.

Рекомендована література:

1. Cohen J. S. Computer Algebra and Symbolic Computation : Elementary Algorithms / J. S. Cohen. Natick : A K Peters, 2002. 1104 p.
2. Петрик М. Р. Диференціальні рівняння у математичних моделях (Моделювання у середовищі Wolfram Mathematica) : навч. посіб. / М. Р. Петрик, І. В. Бойко. Тернопіль : ТНТУ, 2016. 170 с.
3. Maeder R. Computer Science with Mathematica : Theory and Practice for Science, Mathematics, and Engineering / R. Maeder. Cambridge : Cambridge University Press, 2000. 1706 p.
4. Giordano F. R. A First Course in Mathematical Modeling / F. R. Giordano, M. D. Weir, S. B. Horton, W. P. Fox. 4th ed. Belmont : Brooks/Cole, 2009. 418 p.
5. Magrab E. B. Advanced Engineering Mathematics with Mathematica / E. B. Magrab. Boca Raton : CRC Press, 2009. 1271 p.
6. Wellin P. An Introduction to Programming with Mathematica / P. Wellin, R. Gaylord, S. Kamin. 3rd ed. Cambridge : Cambridge University Press, 2019. 804 p.
7. Alves L. Applied Numerical Analysis with Mathematica / L. Alves, M. Mikhailov, R. Cotta. Rio de Janeiro : E-papers Servicos Editoriais, 2017. 646 p.
8. Constanda C. Boundary Integral Equation Methods and Numerical Solutions : Thin Plates on an Elastic Foundation / C. Constanda, D. Doty, W. Hamill. Cham : Springer, 2016. 480 p.
9. Swift R. J. A Course in Ordinary Differential Equations / R. J. Swift, S. A. Wirkus. 2nd ed. Boca Raton : CRC Press, 2011. 504 p.
10. Cole J. D. Perturbation Methods in Applied Mathematics / J. D. Cole. Waltham : Blaisdell Publ. Comp., 1968. 260 p.

11. Prudnikov A. P. Integrals and Series, Vol. 3 : More Special Functions / A. P. Prudnikov, Yu. A. Brychkov, O. I. Marichev. New York : Gordon & Breach Sci. Publ., 1988. 800 p.
12. Gregory P. Bayesian Logical Data Analysis for the Physical Sciences : A Comparative Approach with Mathematica Support / P. Gregory. Cambridge : Cambridge University Press, 2010. 560 p.
13. Uusipaikka E. Confidence Intervals in Generalized Regression Models / E. Uusipaikka. Boca Raton : Chapman & Hall/CRC, 2016. 256 p.
14. Colignatus T. Conquest of the Plane : Using The Economics Pack Applications of Mathematica for a Didactic Primer on Analytic Geometry and Calculus / T. Colignatus. The Hague : Thomas Colignatus, 2016. 312 p.

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 напряму підготовки та надання їм базових знань у студентів із ключових теоретичних і практичних понять та основних перспективних напрямків математичного моделювання та обчислювальних методів. Сформувати 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, 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 символьної 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 різницевими 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. Blower D. J. Information Processing Volume I: Boolean Algebra, Classical Logic, Cellular Automata, and Probability Manipulations / D. J. Blower. [S. l.] : Third Millennium Inferencing, 2017. 564 p.
2. Hastings K. J. Introduction to Probability with Mathematica / K. J. Hastings. 2nd ed. Boca Raton : CRC Press, 2012. 468 p.
3. Самойленко А. М. Диференціальні рівняння у прикладах і задачах / А. М. Самойленко, С. А. Кривошея, М. О. Перестюк. Київ : Вища школа, 2004. 454 с.
4. Самойленко А. М. Диференціальні рівняння / А. М. Самойленко, М. О. Перестюк, М. Ю. Перестюк. Київ : Либідь, 2004. 360 с.
5. Madureira A. L. Numerical Methods and Analysis of Multiscale Problems / A. L. Madureira. Cham : Springer, 2017. 760 p.
6. Петрик М. Основи математичного моделювання в наукових дослідженнях / М. Петрик, М. Баб’юк. Тернопіль : Підручники і посібники, 1998. 120 с.
7. Петрик М. MathCAD-технології в інженерних задачах теорії розрахунку і конструювання / М. Петрик. Тернопіль : Вид-во ТДТУ ім. Ів. Пулюя, 2000. 154 с.
8. Ленюк М. П. Інтегральні перетворення Фур'є, Бесселя із спектральним параметром в задачах математичного моделювання масопереносу в неоднорідних середовищах / М. П. Ленюк, М. Р. Петрик. Київ : Наукова думка, 2000. 372 с.
9. Ляшенко І. М. Моделювання біологічних та екологічних процесів / І. М. Ляшенко, А. П. Мукоєд. Київ : ВПЦ «Київський університет», 2002. 195 с.

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

реалізовують 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:

10. Основні підходи та концепції моделювання ПЗ. Основи проєктування ПЗ. Аналіз Вимог.
11. Основні етапи розробки ПЗ. Життєвий цикл ПЗ.
12. Формалізовані мови моделювання та сучасні засоби проєктування ПЗ.
13. Інструментальні засоби розробки ПЗ. Нові підходи і принципи проєктування ПЗ: "Від вимог до коду".
14. Моделі вимог. Методологія моделювання варіантів використання
15. Ідентифікація класів предметної області на основі Специфікації вимог та описів ВВ. Моделювання класів на UML
16. Моделювання в UML відношень та зв’язків між класифікаторами (класами, ВВ, підсистемами, компонентами).
17. Архітектурний підхід до проєктування діаграм ієрархії класів.
18. Детальне проєктування класів. Розробка деталізованого опису класу на UML та IBM RSA. Моделювання атрибутів та операцій класів
19. Моделювання взаємодій системи з користувачем через використання діаграм взаємодій.
20. Діаграми послідовностей (Sequence Diagram SD). Моделювання повідомлень як взаємодії об'єктів. Діаграми комунікацій (Comunication Diagram)
21. 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.
22. Моделювання потоків керування. Діаграми діяльності.
23. Моделювання об’єктів. Класи і відношення між ними. Структура та проектування складних систем
24. Діаграми станів. Скінченні автомати. Моделювання поведінки спільно працюючих об’єктів у сценаріях варіантів використання.
25. Моделювання інтерфейсів для побудови компонентно-орієнтованої архітектури ПЗ.
26. Моделювання кооперацій, компонентів та розміщення вузлів обладнання.

27. 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.
28. Вибір концептуальної моделі розробки ПЗ на основі аналізу вимог предметної області (згідно варіантів)
29. Аналіз сутностей предметної області як кандидатів в класи системи.
30. Практичні приклади моделей предметної області (МПО). Р реалізація МПО через ідентифікацію класів, що реалізовують усі необхідні для користувача функціональні операції. Приклад моделі глосарію предметної області. Визначення зв'язків між структурними сутностями.
31. 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).
32. 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.
33. 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.
34. Практичні аспекти проектування інтерфейсів (І). І, реалізований компонентою або класом (надаваний). Необхідний інтерфейс компоненти або класу. Розширені зв'язки між компонентою та її інтерфейсами. Заміннність. Приклади компонентів і розміщення для різних предметних областей.
35. Механізми UML-пакетування як засоби вирішення проблеми складності системи. Пакети як підсистеми, що містять компоненти ПЗ. Суть і необхідність компонентної розробки ПЗ. Внутрішня структура пакетів, підсистем компонент.
36. Відповідальність класів . Вимоги проектування класів як елементів архітектури та реалізаторів варіантів використання системи. Розширене розуміння кооперації класів. Класи аналізу і проектування.
37. 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. Імплементація в Java- або C++-код.
38. Системна архітектура програмної системи. Ознаки складної системи

39. Декомпозиція систем. Структура системи. Категорії класів та складові частини об’єктно-орієнтованого підходу.
40. Специфікації. Об’єктна модель
41. 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. UML 2.0 : guide de référence / J. Rumbaugh, I. Jacobson, G. Booch. Paris : Campus Press, 2005. 936 p.
2. Charoux B. UML 2.0 / B. Charoux, A. Osmani, Y. Thierry-Mieg. Paris : Pearson Education France, 2008. 294 p.
3. UML 2.0 Superstructure and UML 2.0 Infrastructure / Object Management Group (OMG). 2004. URL: http://www.uml.org.
4. Gamma E. Design Patterns : Elements of Reusable Object-Oriented Software / E. Gamma, R. Helm, R. Johnson, J. Vlissides. Reading : Addison-Wesley, 1995. 395 p.
5. Hensgen P. Підручник з Umbrello UML Modeller / P. Hensgen, Ю. Чорноіван. 2013. URL: https://docs.kde.org/stable5/uk/kdesdk/umbrello/index.html.
6. Booch G. Object-Oriented Analysis and Design with Applications / G. Booch, I. Jacobson, J. Rumbaugh. 3rd ed. Boston : Addison-Wesley Professional, 2007. URL: 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 / Object Management Group (OMG). 2017. URL: https://www.omg.org/spec/UML/2.5.1/PDF.
8. Табунщик Г. В. Проектування та моделювання програмного забезпечення сучасних інформаційних систем / Г. В. Табунщик, Т. І. Каплієнко, О. А. Петрова. Запоріжжя : ЗНТУ, 2016. URL: http://eir.zntu.edu.ua/bitstream/123456789/1824/1/Tabunshchik_Software_Design.pdf.
9. Dathan B. Object-Oriented Analysis, Design and Implementation : An Integrated Approach / B. Dathan, S. Ramnath. Cham : Springer, 2015. URL: https://warin.ca/ressources/books/2015_Book_ObjectOrientedAnalysisDesignA.pdf.
10. Петрик М. Сучасні технології ООП-проектування та автоматичного генерування програмного коду : навч. посіб. / М. Петрик, І. Мудрик, О. Петрик, Ю. Стоянов. Тернопіль : ТНТУ ім. Івана Пулюя, 2018. 48 с.
11. Петрик М. Моделювання та аналіз програмного забезпечення : наук.-метод. посіб. / М. Петрик, О. Петрик. Тернопіль : Вид-во ТНТУ ім. І. Пулюя, 2014. 180 с.

12. Петрик М. Р. Моделювання програмного забезпечення : наук.-метод. посіб. / М. Р. Петрик, О. Ю. Петрик. Тернопіль : Вид-во ТНТУ ім. Івана Пулюя, 2015. 200 с. URL: http://elartu.tntu.edu.ua/handle/123456789/17796.
13. Rational Software Architect Designer 9.7.0 URL: 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 технології, 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:

14. Поняття об'єктно-орієнтованого програмування. Визначення програмування на основі абстрактних типів даних Інструментальні засоби ООП: інкапсуляція, поліморфізм, наслідування.
15. Загальний формат оголошення класу. Загальний формат визначення методів класу. Різниця між класом, оголошеним зі службовим словом struct і класом, оголошеним зі службовим словом class
16. Що таке об'єкт? Як оголосити обєкт класу? Яка умова є обов'язковою для присвоєння одного об'єкта іншому?
17. Чи можна адресу об'єкта передати у функцію в якості аргументу?
18. Що резервує память оголошення чи визначення класу?
19. Яка область дії членів класу, визначених з атрибутом public?
20. Що означає атрибут private при оголошенні класу?
21. Хто має доступ до даних класу, оголошених з атрибутом protected?

22. Дайте визначення конструктора. Коли він викликається?
23. Що таке деструктор? Коли він викликається?
24. Скільки конструкторів можна оголосити в класі?
25. Скільки деструкторів можна оголосити в класі?
26. Які Ви знаєте види конструкторів?
27. Які Ви знаєте властивості конструкторів?
28. Що таке конструктор копіювання?
29. У яких випадках викликається конструктор копіювання?
30. Що таке конструктор з параметрами? Що таке конструктор по замовчуванню? Сформулюйте основні властивості конструкторів.
31. Який тип можуть мати параметри конструктора?
32. Що таке вказівник this? Яка його функція при ООП?
33. 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?
34. Наведіть приклад використання вказівника this.
35. Як задати параметри для методу по замовчуванню?
36. Дайте визначення дружньої функції.
37. Як оголосити дружню функцію?
38. Дружня функція є методом класу?
39. Чи може дружня функція бути дружньою більше, ніж до одного класу?
40. Чи істинне твердження, що дружня функція має доступ до закритих даних класу, навіть не будучи його методом?
41. Що таке дружній клас?
42. Які Ви знаєте способи перевантаження функцій?
43. Запишіть основну форму унарної оператор-функції методу класу
44. Запишіть основну форму унарної оператор-функції дружньої до класу
45. Запишіть загальний формат бінарної оператор-функції методу класу
46. Запишіть загальний формат бінарної оператор-функції дружньої до класу
47. Які Ви знаєте обмеження на перевантаження операторів?

48. Що таке базовий клас?
49. Що таке похідний клас?
50. Що таке безпосередній базовий клас?
51. Які Ви вивчали приклади поліморфізму?
52. Запишіть формат опису конструктора при множинному наслідуванні
53. Поясніть різницю між простим і множинним наслідуванням.
54. Який атрибут має перевагу: атрибут при наслідуванні чи атрибут у безпосередньому базову класі?
55. Який формат опису класу при простому наслідуванні?
56. Який клас називається абстрактним?
57. Який клас називається поліморфним?
58. У базовому класі встановлений статус доступу public. Яким має бути модифікатор доступу, щоб в похідному класі одержати статус доступу protected?
59. Чи є кожен об'єкт похідного класу об'єктом відповідного базового класу?
60. Чи є кожен об'єкт базового класу об'єктом класів, породжених даним базовим класом?
61. Який формат опису класу при множинному наслідуванні?
62. Поясніть різницю між дружньою, віртуальною і перевантаженою функціями
63. Які Ви знаєте атрибути доступу при наслідуванні?
64. Як відбувається виклик конструкторів при наслідуванні?
65. Як відбувається виклик деструкторів при наслідуванні?
66. Запишіть формат опису конструктора при простому наслідуванні
67. Наведіть приклад ситуації, що приводить необхідності оголошення класу віртуальним.
68. Які Ви знаєте види множинного наслідування? Охарактеризуйте їх (можна схематично).
69. 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()?
70. 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?

71. Наведіть приклад ситуації, що приводить необхідності використання віртуальної функції.
72. Визначення потоку. Класифікація потоків введення / виведення за напрямком передачі даних і за способом створення
73. Схема наслідування класів потоків
74. Засоби форматування потоків виведення
75. Мікросервіси, SOLID принципи, чиста архітектура, DevOps/MLOps, проектування REST API/GraphQL.
76. Форматування потоків введення / виведення за допомогою маніпуляторів
77. Маніпулятори введення, визначені користувачем. Загальний формат
78. Маніпулятори виведення, визначені користувачем. Загальний формат
79. Схема роботи з файлами через потоки
80. Створення потоку введення, виведення, введення-виведення.
81. Функції, що дозволяють зв'язати потік з файлом. Прототип функції open(). Режими відкриття файла через потоки. Методи для запису у файл write і put: прототип, приклад.
82. Методи переміщення поточної позиції введення (виведення) seekp і seekg: прототип, приклад
83. 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?
84. Шаблон функції. Загальна форма визначення шаблону функції
85. Кількість та типи параметрів шаблону функції
86. Шаблон простої функції. Шаблони функцій з параметрами. Вимоги до фактичних параметрів шаблону
87. Основні властивості параметрів шаблона функції. Способи перевантаження шаблонних функцій. Загальна форма визначення шаблону класу. Оголошення обєкта шаблону класу. Шаблони і наслідування. Шаблони і друзі.
88. Поняття механізму виключних ситуацій. Приклади виключень.
89. Випадки, коли необхідно передбачити опрацювання виключних ситуацій

90. Оператори для реалізації механізму обробки виключень. Оператор перехвату виключень try. Оператори обробки виключень catch. Оператори викиду виключень throw. Синтаксис і семантика генерації обробки виключень

91. Введення та виведення в Qt у візуальному режимі.

92. Поняття про класи та об’єкти. Базові принципи ООП.

93. Оголошення класів та об’єктів.

94. Робота з графікою в Qt, координати у графічному режимі. Двовимірна графіка.

95. Відмінності C++ від C.

96. Логічний тип даних у C++. Старий та новий стиль C++. Новий стиль заголовків C++ та простір імен.

97. Оператори введення-виведення <<, >>.

98. Вбудовані потоки у C++. Форматоване введення-виведення C++.

99. Форматування за допомогою членів класу **ios.** Прапорці форматування.

100. Встановлення та скидання прапорців форматування. Ширина поля введення, точність та заповнення.

101. Використання маніпуляторів формату. Маніпулятори формату.

102. Робота з масивом у Qt.

103. Відмінність підпрограм у C та C++. Підпрограми в Qt з використанням ООП.

104. Рядки у C++. Об'єкти класу string. Задавання рядків. Введення та виведення рядків.

105. Рядки у C++. Об'єкти класу string. Операції для роботи з рядками.

106. Рядки у C++. Методи для роботи з рядками. Присвоєння, додавання, порівняння та перетворення рядків.

107. Рядки у C++. Методи для роботи з рядками. Пошук у рядку. Отримання характеристик рядків.

108. Робота з файлами у C++. Потоки для роботи з файлами. Створення файлових потоків.

109. Робота з файлами у C++. Режим відкривання файлів. Відкривання та закривання файлів.

110. Робота з файлами у C++. Читання та запис файлів. Методи читання потоків.

111. Робота з файлами у C++. Читання та запис файлів. Методи запису потоків.

112. Робота у візуальному режимі Qt. Поняття про сигнали та слоти.

113. Робота у візуальному режимі Qt. Підключення сигналів та слотів.

114. Робота у візуальному режимі Qt. Створення своїх сигналів.

115. Робота у візуальному режимі Qt. Створення своїх слотів.

116. Робота у візуальному режимі Qt. Поняття про події та взаємодія з користувачем.

117. Робота у візуальному режимі Qt. Події введення.

118. Робота у візуальному режимі Qt. Перезавантаження методів обробки подій. Події клавіатури.

119. Робота у візуальному режимі Qt. Клас QFocusEvent.

120. Робота у візуальному режимі Qt. Подія оновлення контексту малювання. Події. Події таймера. Таймер.

121. 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 / A. Williams. Shelter Island : Manning Publications Co., 2019. 592 p.
2. Dmitrović S. Modern C++ for Absolute Beginners : A Friendly Introduction to C++ Programming Language and C++11 to C++20 Standards / S. Dmitrović. Berkeley : Apress, 2020. 461 p.
3. Horton I. Beginning Visual C++ 2013 / I. Horton. Indianapolis : Wrox, 2014. URL: https://www.pdfdrive.com/ivor-hortons-beginning-visual-c-2013-d46824353.html.
4. Horton I. Beginning C++17 From Novice to Professional / I. Horton. Berkeley : Apress, 2018. URL: https://www.pdfdrive.com/beginning-c17-from-novice-to-professional-e184792973.html.
5. Stroustrup B. Programming : Principles and Practice Using C++ / B. Stroustrup. Upper Saddle River : Addison-Wesley, 2014. URL: https://www.pdfdrive.com/programming-principles-and-practice-using-c-e18903967.html.
6. Lafore R. Object-Oriented Programming in C++ / R. Lafore. Indianapolis : Sams Publishing, 2002. URL: https://docs.google.com/file/d/0B21HoBq6u9TsUHhqS3JIUmFuamc/view?resourcekey=0-MYlet9RIjEukd6CvLEHUbw.
7. Бублик В. В. Об’єктно-орієнтоване програмування : підручник / В. В. Бублик. Київ : ІТ-книга, 2015. 624 с. URL: https://docs.google.com/file/d/0B21HoBq6u9TsUHhqS3JIUmFuamc/view?resourcekey=0-MYlet9RIjEukd6CvLEHUbw.

8. Петрик О. Об'єктно-орієнтоване програмування в середовищі C++ : навч. посіб. Лабораторний практикум / О. Петрик. Тернопіль : Вид-во ТНТУ ім. Івана Пулюя, 2011. 188 с.
9. Browning B. C++20 Recipes : A Problem-Solution Approach / B. Browning. 2nd ed. Berkeley : Apress, 2020. 657 p.
10. Gregoire M. Professional C++ / M. Gregoire. 5th ed. Indianapolis : Wrox, 2021. 1312 p.
11. Lospinoso J. C++ Crash Course : A Fast-Paced Introduction / J. Lospinoso. San Francisco : No Starch Press, 2019. 792 p.
12. Bancila M. Modern C++ Programming Cookbook / M. Bancila. 2nd ed. Birmingham : Packt Publishing, 2020. 750 p.
13. Solter N. A. Professional C++ / N. A. Solter, S. J. Kleper. 2nd ed. Indianapolis : Wrox, 2022. 1104 p.
15. Swaminathan J. Mastering C++ Programming : Modern C++17/20 Features / J. Swaminathan. Birmingham : Packt Publishing, 2021. 559 p.
16. Sehr V. C++20 Revealed : Discover the Most Modern, Effective, and Safe Programming Language / V. Sehr. Berkeley : Apress, 2020. 547 p.
17. Object Oriented Programming (OOP) in C++ Course // freeCodeCamp.org. 2020. URL: https://www.youtube.com/watch?v=wN0x9eZLix4.
18. Concepts of OOP // Darshan University. URL: https://www.youtube.com/watch?v=hyJeuFWmROc&list=PLftJ4X48yC1k97-tmIpp0HpW-lh2dv9i2.
19. Standard Template Library (STL) // Topcoder. URL: http://community.topcoder.com/tc?module=Static&d1=tutorials&d2=standardTemplate Library.
20. Стандартна бібліотека. Довідник. URL: http://www.cplusplus.com/reference/.
21. Введення в об'єктно-орієнтоване програмування // CPP Studio. URL: http://cppstudio.com/uk/cat/274/281/.
22. Довідник по C++ // Microsoft Developer Network (MSDN). URL: https://msdn.microsoft.com/uk-ua/library/3bstk3k5.aspx.
23. Система онлайн-тестування // Quizful. URL: http://www.quizful.net/category/cpp.
24. Сайт з найновіщою довідкою по C++ URL: http://www.cplusplus.com.
25. Visual Studio Team Services // Microsoft Learn. URL: https://learn.microsoft.com/uk-ua/.
26. IDE Visual C++
27. // Microsoft Support. URL: https://support.microsoft.com/uk-ua/help/2977003/the-latest-supported-visualc-downloads.

7. НАУКОВЕ МИСЛЕННЯ ТА ДОСЛІДНИЦЬКА СПРОМОЖНІСТЬ

1. Основні поняття дисертаційного дослідження.
2. Ключові складові дисертаційної роботи.
3. Методи наукового дослідження в інженерії ПЗ: формальні методи, теоретичні підходи та емпірична програмна інженерія (контрольовані експерименти, кейс-стаді, опитування).
4. Академічна доброчесність, цитування, плагіат, коректність оформлення літературних джерел.
5. Поняття наукової публікації, апробація результатів, наукометрія та рецензування.
6. Аналіз складності, оптимізація та рефакторинг програмних рішень. Метрики оцінювання якості вихідного коду та архітектури ПЗ.
7. Штучний інтелект в інженерії ПЗ (AI4SE) та інженерія ПЗ для систем ШІ (SE4AI). Автоматизована генерація коду, пошук дефектів та особливості тестування ML-інтенсивних програмних систем.
8. Аналітика програмного забезпечення (Software Analytics) та методи дослідження програмних репозиторіїв (Mining Software Repositories — MSR).
9. Архітектурне та компонентне моделювання складних програмних систем. Патерни проектування, проектування розподілених та хмарних застосунків.
10. Методології верифікації та валідації ПЗ. Формальна специфікація програмних систем, автоматизація тестування та управління процесами розгортання (DevOps/MLOps інфраструктури).
11. Алгоритми, обчислювальна складність та оптимізація програмних рішень.
12. Машинне навчання, нейронні мережі та оцінювання якості моделей.
13. Аналіз даних, Data Science та технології Big Data.

III. ПОРЯДОК ПРОВЕДЕННЯ ТА КРИТЕРІЇ ОЦІНЮВАННЯ ІСПИТУ ЗІ СПЕЦІАЛЬНОСТІ

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

Структура тестових завдань передбачає 7 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**

IV. ОСКАРЖЕННЯ РЕЗУЛЬТАТІВ ІСПИТУ ЗІ СПЕЦІАЛЬНОСТІ

Вступник має право на подання апеляції на результат вступного іспиту зі спеціальності, 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.