

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Чорноморський національний університет
імені Петра Могили

ЗАТВЕРДЖЕНО

Ректор ЧНУ імені Петра Могили



Леонід КЛИМЕНКО

2023 р.

ПРОГРАМА
фахового іспиту
для вступу на 1 курс навчання
другого (магістерського) рівня вищої освіти
для здобуття ступеня магістра
зі спеціальності
«121 Інженерія програмного забезпечення»

Миколаїв - 2023

ЗМІСТ

1 ПОЯСНЮВАЛЬНА ЗАПИСКА	2
2 СТРУКТУРА ТЕСТОВОГО ЗАВДАННЯ	5
3 КРИТЕРІЇ ОЦІНЮВАННЯ ФАХОВОГО ВСТУПНОГО ВИПРОБУВАННЯ	6
4 ТЕМАТИЧНИЙ ЗМІСТ ДЛЯ ПІДГОТОВКИ ДО ФАХОВОГО ВСТУПНОГО ВИПРОБУВАННЯ	7
5 НАВЧАЛЬНО-МЕТОДИЧНІ МАТЕРІАЛИ	21

1 ПОЯСНЮВАЛЬНА ЗАПИСКА

Прийом на підготовку фахівців освітньо-кваліфікаційного рівня магістр за спеціальністю 121 «Інженерія програмного забезпечення» на 1-й курс здійснюється на основі здобутого освітньо-кваліфікаційного рівня бакалавр за спеціальністю 121 «Інженерія програмного забезпечення» або споріднених спеціальностей.

Фахове вступне випробування має вигляд іспиту, що триває 1 годину та охоплює 6 освітніх компонент: «Об'єктно-орієнтоване програмування», «Організація баз даних», «Алгоритми та структури даних», «Web development», «Операційні системи», «Якість програмного забезпечення та тестування».

Метою фахового вступного випробування з дисципліни **«Об'єктно-орієнтоване програмування»** є перевірка знань абітурієнтів про:

- поняття конструкторів та дедукторів;
- основні типи наслідування;
- властивості похідних класів;
- призначення `override` та `final`;
- віртуальну функцію та абстрактні класи;
- механізм пізнього зв'язування;
- статичний та динамічний поліморфізм;
- типи поліморфізму;
- інкапсуляцію та способи обмеження доступу до членів класу;
- базові оператори Java;
- робота з класами та функціями Java;
- об'єктно-орієнтоване програмування з Java.

Метою фахового вступного випробування з дисципліни **«Організація баз даних»** є перевірка знань абітурієнтів про:

- проєктування баз даних, моделювання із застосуванням IDF- та DFD-методологій;
- реляційні таблиці;

- шаблони даних;
- запити в SQL;
- реляційну модель бази даних;
- проєктування логічної організації даних: стандарти DFD та ERD;
- команди маніпулювання даними;
- команди управління даними.

Метою фахового вступного випробування з дисципліни **«Алгоритми та структури даних»** є перевірка знань абітурієнтів про:

- призначення та властивості основних структур даних;
- фізичну та логічну структуру найбільш поширених структур даних;
- типові алгоритми роботи з найбільш поширеними структурами даних;
- методи реалізації найбільш поширених структур даних;
- алгоритми сортування;
- файлові структури даних;
- напівстатичні структури даних з послідовною організацією доступу;
- динамічні структури даних.

Метою фахового вступного випробування з дисципліни **«Web development»** є перевірка знань абітурієнтів про:

- використання фреймворків при розробці web-застосунків;
- поняття менеджера пакетів;
- наслідування, перевизначення та передача даних;
- створення контролеру та маршрутизацію із використанням методів контролеру;
- валідацію даних та обробку помилок;
- взаємодію із базою даних через класи моделі Eloquent;
- поняття міграції;
- автентифікацію та авторизацію користувачів;
- систему контролю прав доступу.

Метою фахового вступного випробування з дисципліни **«Операційні системи»** є перевірка знань абітурієнтів про:

- поняття операційної системи;
- командну оболонку в ОС Linux;
- файлову систему Linux;
- основні операції для роботи з файлами та каталогами;
- стандартні потоки даних у Linux та їх перенаправлення;
- використання текстових редакторів у консольному режимі ОС Linux;
- права доступу для файлів та каталогів Linux, їх встановлення;
- роботу з текстовими файлами;
- багатозадачність у Linux;
- автоматизацію виконання регулярних завдань;
- програмування функцій у сценаріях bash;
- команди для роботи з мережею у Linux;
- основні можливості редактору sed;
- багатопотоковість у Linux.

Метою фахового вступного випробування з дисципліни **«Якість програмного забезпечення та тестування»** є перевірка знань абітурієнтів про:

- загальні поняття, визначення та концептуальні положення щодо тестування;
- складання вимог до програмного продукту;
- організацію та оформлення процесу тестування;
- задачі та цілі складання тестових сценаріїв (test case);
- техніки тестування методом «Чорної та білої скриньки»;
- модульне та інтеграційне тестування;
- автоматизоване тестування;
- використання Jmeter для функціонального та навантаженого тестування;
- особливості тестування вебзастосунків та мобільних застосунків.

2 СТРУКТУРА ТЕСТОВОГО ЗАВДАННЯ

Тестове завдання складається із 30 питань, що охоплюють усі 6 дисциплін. Структуру білету та розподіл питань по дисциплінах наведено у таблиці 1.

Таблиця 1 – Структура білету та розподіл питань по дисциплінах

№	Назва дисципліни з навчального плану	Обсяг за рік, кред.	Кількість питань
1.	Об'єктно-орієнтоване програмування	9	5
2.	Організація баз даних	5	5
3.	Алгоритми та структури даних	4	5
4.	Web development	4	5
5.	Операційні системи	5	5
6.	Якість програмного забезпечення та тестування	5	5
	РАЗОМ	32	30

До завдання включено тести з варіантами відповідей, серед яких треба вказати 1 правильну, де інші усі неправильні, або 1 неправильну, де інші усі правильні, відповідь.

3 КРИТЕРІЇ ОЦІНЮВАННЯ ФАХОВОГО ВСТУПНОГО ВИПРОБУВАННЯ

Екзаменаційне завдання містить 30 тестових питань, що охоплюють всі теми, наведені в тематичному змісті даної програми. Таким чином, правильна відповідь на 30 тестових питань оцінюється у 200 балів.

Таблиця 2 – Розподіл балів за кожне завдання

Тестовий бал	Бал за шкалою 100–200
3	100
4	107
5	114
6	121
7	126
8	131
9	134
10	137
11	140
12	143
13	145
14	147
15	148
16	150
17	151
18	152

Тестовий бал	Бал за шкалою 100–200
19	153
20	155
21	157
22	159
23	163
24	167
25	171
26	175
27	181
28	187
29	193
30	200

На підставі виконання фахового вступного випробування комісія оцінює знання та вміння абітурієнта і ухвалює рішення про прийом абітурієнта для навчання на спеціальність 121 «Інженерія програмного забезпечення» або відмовляє в прийомі.

4 ТЕМАТИЧНИЙ ЗМІСТ ДЛЯ ПІДГОТОВКИ ДО ФАХОВОГО ВСТУПНОГО ВИПРОБУВАННЯ

Питання для самоконтролю з дисципліни «Об'єктно-орієнтоване програмування»

1. Що таке ООП?
2. Які переваги використання ООП?
3. Які недоліки використання ООП?
4. Що таке константа, та чим відрізняється від змінної оголошеної як readonly в C#?
5. Що означає рядок незмінний в C#?
6. Які властивості перевантажених операцій в C# ви знаєте?
7. Перелічіть атрибути серіалізації в C#.
8. Перелічіть типи серіалізації в C#.
9. Які бібліотечні інтерфейси Ви знаєте в C#?
10. Перелічіть атрибути серіалізації в C#?
11. Що таке делегат в C#?
12. Які вбудовані делегати в C# Ви знаєте?
13. Наведіть приклади використання вбудованих делегатів в C#.
14. Як працює оператор циклу for?
15. Як працює оператор циклу while?
16. Як працює оператор циклу do-while?
17. Що таке умовний оператор?
18. Що таке безумовний оператор?
19. Оператор перемикач?
20. Як працює оператор перемикач?
21. Що таке масив?
22. Що таке багатовимірний масив?
23. Що таке структура?
24. Що таке перерахування?
25. Що таке перевантажені методи?
26. Що таке перевизначені методи?
27. Що таке клас?
28. Перелічіть модифікатори доступу в C#?
29. Що таке властивості в C#?
30. Що таке індексатори в C#?
31. Що таке конструктор?
32. Які властивості конструкторів ви знаєте?
33. Що таке автоматичні властивості в C#?
34. Що не наслідуються екземпляром похідного класу від базового?
35. Що таке механізм пізнього зв'язування?
36. Що таке віртуальний метод в C#??
37. Для чого існують статичні дані та статичні методи в C#?
38. Статичні класи в C#.

39. Структура java-класу.
40. Що таке конструктор, його призначення та властивості.
41. Типи даних.
42. Що таке сигнатура методу.
43. Що таке перегрузка та перевизначення методів
44. Особливості роботи конструкторів при спадкуванні.
45. Модифікатори доступу — особливості використання.
46. Використання ключового слова `final` до атрибутів, методів та класів.
47. Локальні змінні.
48. Синтаксис звернення до атрибутів класу та його методів .
49. Статичні типи атрибутів класу та статичні методи.
50. В яких випадках користуються ключовим словом `super`.
51. Що означає ключове слово `extends` у класі, як воно відображається н властивостях класу.
 52. Які елементи класу можуть бути пов'язані з інкапсуляцією, за допомогою яких ключових слів це відбувається.
 53. Що таке інтерфейс. Як пов'язані інтерфейси з поняттям успадкування.
 54. Основні властивості інтерфейсних змінних.
 55. Види ініціалізації масивів.
 56. У який спосіб можна встановити кількість рядків та строк у багатовимірних масивах.
 57. Перерахуйте основні (не менш 7) методи класу `Arrays` та їх призначення і синтаксис використання.
 58. Типові методи перетворення типів даних.
 59. Чим відрізняється тип `Float` від `float`.
 60. Успадкування у інтерфейсів.
 61. Перерахуйте усі конструкції, які можуть використовуватися при перехопленні виключень.
 62. Що таке виключення. Види виключень
 63. Які можуть бути варіанти обробки виключень.
 64. Перерахуйте методи кореневого класу всіх класів `Java`.
 65. Чим відрізняється `finalized()` від `finally`. Коротка характеристика кожного елементу.
 66. Архітектура `Model-View-Controller`
 67. Модель обробки подій у графічному інтерфейсі користувача.
 68. Які менеджери розміщення ви знаєте.
 69. Чим відрізняється конструкції `while` та `do while`.
 70. Особливості використання конструкції `case`.
 71. Дві форми використання оператора `for`.
 72. Синтаксис умовного оператора, особливості використання.
 73. Як пов'язані `AWT` та `Swing`.
 74. Які обов'язкові класи повинні бути в коді графічного інтерфейсу.
 75. Як анонімні та внутрішні класи використовуються при організації графічного інтерфейсу.
 76. Основні інтерфейси, які використовуються для роботи з базами даних

77. Структура програми для роботи з БД.
78. Методи, які використовуються для програмної модифікації даних в БД.
79. Виключення, які можуть виникати при роботі з БД.
80. Переваги використання колекцій.
81. Ключові властивості основних типів колекцій.
82. Основні інтерфейси колекцій.
83. Як пов'язані інтерфейси Set та Comparable.
84. Чим відрізняється інтерфейси ListIterator та Iterator.
85. Перерахуйте основні методи інтерфейсу Collection.
86. Що таке абстрактний клас, в якому разі клас може бути абстрактним.
87. Поняття потоку, як отримати доступ до свойств потоку.
88. Керування потоком.
89. Типові способи вирішення конфліктів при конкуренції потоків за ресурси.
90. Чим відрізняються поняття серіалізації та синхронізації.
91. Поняття семафорів.
92. Як організувати пул потоків.
93. Як передати дані з одного потоку у другий.
94. Стани, в яких може знаходитись потік.
95. 2 способи створення потоків.

Питання для самоконтролю з дисципліни «Організація баз даних»

1. Дайте визначення терміну «атрибут».
2. Дайте визначення терміну «кортеж».
3. Дайте визначення терміну «таблиця».
4. Дайте визначення терміну «домен».
5. Дайте визначення терміну «ключ таблиці».
6. Дайте визначення терміну «база даних».
7. Взаємозв'язок понять «таблиця» і «файл».
8. Взаємозв'язок понять «кортеж», «рядок» і «запис».
9. Взаємозв'язок понять «атрибут», «стовпчик таблиці» і «поле запису».
10. Різниця між базою даних і базою знань.
11. Дайте визначення терміну «плоский файл».
12. Коли можна застосовувати теорію множин стосовно плоских файлів.
13. Недоліки файлової організації даних.
14. Переваги файлової організації даних.
15. Основні операції з файлами.
16. Програмні засоби ведення файлів.
17. Ранні підходи до побудови баз даних.
18. Що таке «система керування базами даних»?
19. Основні функції СКБД.
20. Основні функції адміністратора баз даних.
21. Якими характеристиками визначається модель даних?
22. Якими структурними елементами оперує ієрархічна модель даних?

23. Якими структурними елементами оперує мережева модель даних?
24. Яким чином задаються зв'язки між структурними елементами в ієрархічній моделі даних?
25. Яким чином задаються зв'язки між структурними елементами в мережевій моделі даних?
26. Охарактеризуйте тип мови пошуку та маніпулювання даними в ієрархічній моделі даних.
27. Охарактеризуйте тип мови пошуку та маніпулювання даними в мережній моделі даних.
28. Як забезпечується посилальна цілісність в ієрархічній моделі даних.
29. Як забезпечується посилальна цілісність в мережній моделі даних.
30. У чому полягає основна різниця між ієрархічною та мережевою моделями даних.
31. Дайте визначення ієрархічної моделі даних.
32. Дайте визначення мережної моделі даних.
33. Недоліки і переваги ієрархічної моделі даних.
34. Недоліки і переваги мережної моделі даних.
35. Дайте визначення операції «декартів добуток» двох множин в теорії множин.
36. Дайте визначення операції «відношення» двох множин в теорії множин.
37. Чим будуть відрізнятися результати операцій декартового добутку двох множин і бінарного відношення між тими ж множинами?
38. Чим наглядно можна представити відношення?
39. Дайте визначення терміну «відношення» між n множинами (базовими відношеннями).
40. Дайте визначення реляційної моделі даних.
41. Основні вимоги до реляційної моделі даних.
42. Назвіть вимоги до двовимірної таблиці, що її можна було вважати відношенням?
43. Назвіть властивості відношень.
44. Що таке «кортеж» і що таке «атрибут» відношення?
45. Що таке «кардинальне число» і що таке «ступінь» відношення?
46. Як задаються зв'язки між структурними елементами в реляційній моделі даних?
47. На чому ґрунтуються мови маніпулювання структурними елементами у реляційній моделі даних?
48. У чому полягає принципова відмінність реляційної моделі від ієрархічної та мережевої?
49. З яких операцій складається початкова реляційної алгебра Кодда?
50. Дайте визначення операцій РА «переріз».
51. Дайте визначення операцій РА «різниця».
52. Дайте визначення операцій РА «об'єднання».
53. Яким вимогам мають відповідати відношення, щоб до них можна було застосовувати операції «переріз», «різниця» і «об'єднання».

54. Дайте визначення операцій РА «прямий добуток».
55. Дайте визначення операцій РА «проекція».
56. Дайте визначення операцій РА «вибірка».
57. Дайте визначення операцій РА «ділення».
58. Дайте визначення операцій РА «природне з'єднання».
59. Дайте визначення операцій РА «загальне з'єднання».
60. Дайте визначення операцій РА «напівпереріз».
61. Дайте визначення операцій РА «напіврізниця».
62. Дайте визначення операцій РА «ліве з'єднання».
63. Дайте визначення операцій РА «праве з'єднання».
64. Дайте визначення операцій РА «групування з одночасним застосуванням агрегатних функцій».
65. Дайте визначення операцій РА «перейменування атрибута(ів)».
66. Дайте визначення операцій РА «додавання атрибуту».
67. Дайте визначення операцій РА «видалення атрибуту».
68. Як будуються вирази в реляційній алгебрі.
69. Яка різниця між реляційним численням та реляційною алгеброю?
70. Які існують варіанти реляційного числення?
71. Що таке «змінна кортежів»?
72. Які змінні в реляційному численні називають вільними? 73 6 5 Які змінні в реляційному численні називають зв'язаними?
73. Правильно побудовані формули реляційного числення (WFF).
74. Використання квантора загальності.
75. Використання квантора існування.
76. Список цільових елементів.
77. Дайте порівняння обчислювальних можливостей реляційного числення і
 78. реляційної алгебри
 79. Мова QBE. Як задаються таблиці та їх атрибути?
 80. Мова QBE. Створення індексів і ключів.
 81. Мова QBE. Як подаються таблиці і запити на аркуш запитів?
 82. Мова QBE. Як подаються таблиці і запити на аркуш запитів?
 83. Мова QBE. Як подаються атрибути на аркуш запитів?
 84. Як задаються умови вибірки в запитах?
 85. Мова QBE. Як задається умова вибірки з використанням операції «або» в умові вибірки?
 86. Мова QBE. Як задається умова вибірки з використанням операції «й» в умові вибірки?
 87. Які типи запитів передбачені в мові QBE?
 88. Мова QBE. Як створюється перехресна таблиця?
 89. Чи є в мові QBE елементи реляційної алгебри?
 90. Чи є в мові QBE елементи реляційного числення?
 91. Чи є в мові SQL елементи реляційної алгебри?
 92. Чи є в мові SQL елементи реляційного числення?

93. Мова SQL. Наведіть граматику, семантику і приклад оператора CREATE TABLE
94. Мова SQL. Визначення ключа таблиці.
95. Мова SQL. Визначення індекс(ів) таблиці.
96. Мова SQL. Наведіть граматику, семантику і приклад операторів видалення таблиці і видалення індексів.
97. Мова SQL. Наведіть граматику, семантику і приклад операторів видалення атрибутів з таблиці.
98. Мова SQL. Наведіть граматику, семантику і приклад оператора додавання атрибутів до таблиці.

Питання для самоконтролю з дисципліни «Алгоритми та структури даних»

1. Основні визначення згідно Міжнародних термінологічних стандартів (ISO 2382-1:1993, IEEE Std 829-2008): програмне забезпечення, програма, дані, структура.
2. Сутність поняття «структура даних».
3. Класифікація структур даних за рівнем складності.
4. Класифікація структур даних за наявністю зав'язків між елементами структури.
5. Класифікація структур даних за мінливістю. Загальна характеристика.
6. Класифікація структур даних за мінливістю. Статичні структури. Приклади використання.
6. Класифікація структур даних за мінливістю. Напівстатичні структури. Приклади використання.
7. Класифікація структур даних за мінливістю. Динамічні структури. Приклади використання.
8. Класифікація структур даних за характером впорядкованості її елементів.
9. Класифікація структур даних за розміщенням в пам'яті ЕОМ.
10. Характеристика структури – зв'язані списки. Приклади задач, у яких використовується ця структура.
11. Характеристика структури – вектор. Приклади задач, у яких використовується ця структура.
12. Характеристика структури – масив. Приклади задач, у яких використовується ця структура.
13. Характеристика структури – запис (в мові C++ це структура). Приклади задач, у яких використовується ця структура.
14. Характеристика структури – стек. Приклади задач, у яких використовується ця структура.
15. Характеристика структури – двобічної черги. Приклади задач, у яких використовується ця структура.
16. Характеристика структури – черга. Приклади задач, у яких використовується ця структура.

17. Характеристика структури – рядок. Приклади задач, у яких використовується ця структура.
18. Характеристика структури – дерева. Приклади задач, у яких використовується ця структура.
19. Характеристика структури – бінарне дерево. Приклади задач, у яких використовується ця структура.
20. Характеристика структури – дерево пошуку. Приклади задач, у яких використовується ця структура.
21. Характеристика структури – графи. Приклади задач, у яких використовується ця структура.
22. Методи сортування масивів. Загальна характеристика.

Питання для самоконтролю з дисципліни «Web development»

1. Поняття фреймворку розробки.
2. Огляд фреймворків розробки програмного забезпечення на мові PHP.
3. Фреймворк Laravel.
4. Поняття менеджера пакетів.
5. Менеджер пакетів Composer.
6. Поняття request та response в web-програмуванні.
7. Організація маршрутизації у Laravel.
8. Використання шаблонізаторів при розробці представлень користувача.
9. Шаблонізатор Blade.
10. Використання ORM при розробці web-застосунків.
11. ORM Eloquent та класи моделі.
12. Використання контролерів у Laravel.
13. Створення представлення за допомогою шаблонізатору Blade у laravel.
14. Повернення представлення користувачу у контролері або функції зворотнього виклику.
15. Організація маршрутизації та web.php.
16. Наслідування шаблонів Blade.
17. Конструкція @extends шаблонізатору Blade.
18. Перевизначення секцій у шаблонах Blade.
19. Конструкція @yield шаблонізатору Blade.
20. Конструкція @section шаблонізатору Blade.
21. Передача даних до шаблону Blade.
22. Методи with та withVariable.
23. Виведення даних за допомогою Blade. Конструкція {{ }}.
24. Екратування виводу у Blade. Конструкція {!! !!}.
25. Організація @foreach у шаблонах Blade.
26. Організація маршрутизації із використанням контролерів.
27. Підключення css-стилів до проекту Laravel.
28. Підключення js-бібліотек до проекту Laravel.
29. Html-форми.

31. Поняття Cross-Site Request Forgery.
32. Засоби уникнення Cross-Site Request Forgery.
33. Конструкція `csrf_field()`.
34. `Blade @csrf_field`.
35. Методи запитів користувача для додавання, модифікації та видалення даних.
36. Конструкція `method_field()`.
37. `Blade @method`
38. Функція `dump and die`
39. Організація аґах-запитів.
40. Передача `_method` та `_token` в аґах-запиті.
41. Реалізація контролеру ресурсів.
42. Поняття Laravel route model binding.
43. Поняття валідації даних.
44. Два прошарки валідації.
45. Валідація на клієнті засобами інтернет-браузеру.
46. Типи обмежень даних.
47. Недоліки клієнтської валідації.
48. Засоби Laravel для реалізації валідації даних на сервері.
49. Повернення даних методом `validate()`.
50. Обробка та виведення помилок валідації засобами шаблонізатору Blade.
51. Виведення на формі старих даних після невдалої валідації даних.
52. Поняття ORM Eloquent.
53. Міґрації ORM Eloquent.
54. Запуск та відкат міґрації.
55. Методи `up()` та `down()` класу міґрації.
56. Типи зв'язків між таблицями БД.
57. Створення поля та зовнішнього ключа за допомогою міґрації.
58. Конструкція `belongsTo()`.
59. Конструкція `hasOne()`.
60. Конструкція `hasMany()`.
61. Звернення до головного об'єкту через відповідний метод класу залежної моделі.
62. Захищені масиви `$fillable` та `$guarded` класу моделі.
63. Оновлення даних зовнішнього ключа за значенням `id`.
64. Оновлення даних зовнішнього ключа із використанням конструкції `associate()`.
65. Конструкція `@include` шаблонізатора `@blade`.
66. Реалізація типу зв'язку один до багатьох в Eloquent.
67. Визначення відношення у залежному класі моделі.
68. Визначення відношення у головному класі моделі.
69. Отримання колекції екземплярів класу залежної моделі із екземпляру головної моделі.
70. Конструкції `belongsTo()` та `hasMany()`.

71. Реалізація маршрутизації із вказанням індексу головної моделі.
72. Визначення конструктору контролера.
73. Визначення глобальних змінних представлень.
74. Валідація зовнішнього ключа.
75. Отримання списку залежних об'єктів через клас моделі головного об'єкту.
76. Автентифікація та авторизація користувачів у Laravel.
77. Конфігурування бази даних та налаштування автентифікації.
78. Маршрутизація автентифікації користувачів.
79. Ручна аутентифікація
80. Збереження користувача та отримання його даних.
81. Контроль автентифікованості при доступі до ресурсів.
82. Gates.
83. Policies.
84. Робота через middleware.
85. Робота через модель.
86. Робота через контролер.
87. Поняття сервіс-контейнеру.
88. Поняття сервіс-провайдеру.
89. Поняття Middleware.
90. Робота із електронною поштою.
91. Повідомлення користувачу, сесії та моментальні повідомлення.

Питання для самоконтролю з дисципліни «Операційні системи»

1. Поняття «операційна система», її призначення та місце в обчислювальних системах. Основні характеристики операційних систем.
2. Поняття вільного програмного забезпечення (ПЗ) і ПЗ з відкритим вихідним кодом. Приклади вільних операційних систем.
3. Основні особливості і огляд операційних систем сімейства UNIX. Особливості ОС Linux.
4. Поняття «дистрибутив» ОС Linux, основні характеристики. Основні дистрибутиви ОС Linux.
5. Структура системи каталогів ОС Linux. Основні системні каталоги у Linux. Різновиди маршрутних імен файлів. Приклади.
6. Команди управління файлами в ОС Linux. Приклади застосування.
7. Види посилань на файли, їх відмінності. Команда створення посилань на файли. Приклади.
8. Поняття файлу. Типи файлів. Атрибути файлів. Команди виведення атрибутів файлів і пошуку файлів по атрибутам.
9. Перенаправлення введення, виведення та помилок. Призначення. Приклади використання.

10. Пошук по вмісту файлу за заданим шаблоном. Використання групових символів та регулярних виразів при пошуку за шаблоном. Приклади.

11. Можливості команди конкатенації і виведення вмісту файлів. Приклади застосування.

12. Використання конвеєрів та фільтрів. Принципи роботи. Приклади використання.

13. Робота у текстових редакторах vi та nano. Режими редактора vi та перехід між ними. Основні команди редактора nano.

14. Основні характеристики облікового запису користувача. Структура та формат файлу /etc/passwd. Команди виведення інформації про дані облікового запису користувача.

15. Права доступу до файлів і каталогів в ОС Linux. Кодування прав доступу. Вплив прав доступу на виконання операцій над файлами і каталогами.

16. Поняття власника і групи-власника файлу в ОС Linux. Команди зміни атрибутів файлу.

17. Команди управління правами доступу до файлів і каталогів. Права за замовчуванням. Приклади.

18. Команди порівняння файлів. Формат виведення результатів команд. Просте та детальне порівняння. Застосування результатів порівняння для зміни вмісту файлів.

19. Сортування файлів. Режими сортування. Сортування за полями, використання розділювачів полів. Сортування по декільком полям та певній частині поля.

20. Вибірка окремих полів з рядків файлу. Вибірка за полями та за символами. Команди склеювання файлів.

21. Пошук повторюваних рядків у файлі. Особливості використання та основні режими. Приклади.

22. Команди пошуку файлів в системі каталогів. Принципи використання та відмінності. Приклади.

23. Використання перевірок при пошуку файлів. Перевірка за типом, розміром, іменем файлу. Об'єднання декількох перевірок.

24. Команди стискання файлів. Основні режими роботи. Повернення файлу до початкового розміру. Ступені стиснення файлів. Команди перегляду стиснених файлів.

25. Команди архівації файлів, їх відмінності. Основні режими архівації файлів: створення архіву, перегляд його вмісту, додавання нового файлу до існуючого архіву, розпакування архіву.

26. Поняття процесу у ОС Linux. Загальна схема організації процесів. Дескриптор процесу. Засоби взаємодії процесів.

27. Засоби командного рядка для отримання інформації про процеси. Відомості про процеси.

28. Принципи організації багатозадачного режиму виконання завдань з терміналу користувача. Виведення списку завдань. Переведення завдання у фоновий режим та його повернення на передній план. Призупинення процесу.

29. Посилання сигналів процесам у ОС Linux, варіанти використання. Основні сигнали. Посилання сигналів декільком процесам.

30. Отримання інформації про стан системи та її основні розділи. Інформація про об'єм вільного простору на окремих пристроях та каталогах.

31. Планування регулярного виконання завдань для допомогою cron-таблиць, їх формат. Управління планувальником.

32. Призначення і функції оболонки bash. Загальна структура файлу-сценарію командної оболонки. Способи запуску сценаріїв.

33. Середовище оточення користувача і засоби його налаштування. Виведення змінних середовища оточення. Додавання нової змінної до середовища оточення. Зміна рядка-запрошення до введення команд.

34. Оголошення змінних у сценаріях bash. Типи змінних. Присвоєння та виведення значень змінних. Видалення змінної. Оголошення типізованої змінної. Введення значення змінної користувачем.

35. Оголошення масивів у bash, їх ініціалізація. Виведення певного елементу масиву та всього масиву в цілому.

36. Складені команди у bash. Статус завершення виконання команди. Повернення зі сценарію статусу виходу.

37. Поняття екранування у bash, його засоби. Конкатенація рядків та виведення кількості символів у змінній.

38. Підстановка значень змінних та результатів роботи команд у bash. Приклад.

39. Обчислення виразів у bash. Основні арифметичні оператори. Приклади.

40. Передача та обробка параметрів у сценаріях bash. Використання спеціальних змінних командної оболонки. Команди зсуву та присвоєння значень параметрів командного рядка у сценаріях.

41. Команда розгалуження у сценаріях bash, її особливість у порівнянні з іншими мовами програмування. Синтаксис команди. Приклади.

42. Команда перевірки умов у сценарії bash. Перевірка файлів, рядків та чисел. Приклади.

43. Команда вибору довільної кількості альтернатив у bash. Семантика конструкції. Приклади.

44. Використання циклів `while` та `until` у сценаріях `bash`. Синтаксис. Приклади.
45. Використання циклів `for` у сценаріях `bash`. Різновиди циклу `for`. Синтаксис. Приклади.
46. Управління циклами у сценаріях `bash`. Використання вкладених циклів. Вихід з внутрішнього та зовнішнього циклів. Приклади.
47. Створення меню у сценаріях `bash` за допомогою циклу `select`. Синтаксис. Приклади.
48. Синтаксис оголошення функції у сценаріях `bash`. Виклик функції у сценарії. Можливі помилки при визначенні та виклику функцій. Передача параметрів у функцію.
49. Способи повернення значення з функції у сценаріях `bash`. Приклади.
50. Обробка змінних у функціях у сценаріях `bash`. Локальні та глобальні змінні. Приклади.
51. Передача та повернення масивів з функцій у сценаріях `bash`. Приклади.
52. Створення бібліотек функцій `bash`. Підключення бібліотеки у сценарії `bash`. Приклади.
53. Система `X Window System`. Налаштування `X`-сервера та `X`-клієнта. Емулятор терміналу `xterm`. Основні графічні середовища `Linux`.
54. Використання діалогових вікон для створення сценаріїв `bash` для роботи у графічному середовищі. Пакети `kdiallog` та `zenity`, їх основні складові.
55. Основні команди дослідження та моніторингу мережі у `Linux`.
56. Команди передачі файлів у мережі та входу до віддалених комп'ютерів у `Linux`.
57. Використання регулярних виразів при пошуку тексту у `Linux`. Метасимволи, що використовуються у регулярних виразах.
58. Класи символів у регулярних виразах `Linux`. Приклади.
59. Прості та розширені регулярні вирази у `POSIX`. Додаткові метасимволи у розширених регулярних виразах.
60. Збіг елементів повну кількість разів у розширених регулярних виразах у `Linux`.
61. Використання компіляторів `gcc` та `g++` для розробки програм на мовах `C` та `C++` у `Linux`. Стадії побудови програми. Приклади.
62. Створення проектів з декількох файлів `C/C++` у компіляторах `gcc/g++`. Підключення бібліотек до проєкту. Приклади.
63. Використання утиліти `make` для зборки програм на мовах `C/C++`. Формат `make`-файлів. Приклади.

Питання для самоконтролю з дисципліни «Якість програмного забезпечення та тестування»

1. Якість ПЗ. Моделі якості ПЗ. Зв'язок тестування з якістю ПЗ.
2. Тестування методом «білого ящика» – критерії покриття. Техніки тестування.
3. Міжнародні стандарти , які пов'язані з тестуванням.
4. Види тестування по SWEBOOK
5. Статичне тестування. Інспекції, аудити, технічні оцінки.
6. Баг- визначення, види. Життєвий цикл бага по Bugzilla.
7. Статичне тестування і GRASP-патерни. Перерахувати і коротко описати всі 9 патернів.
8. Техніки тестування методом «чорного ящика» – аналіз граничних значень, розбиття вхідних даних на класи еквівалентності, таблиця прийняття рішень.
9. Техніки тестування методом «чорного ящика» – State Transition testing.
10. Тестування зручності використання (основні поняття, особливості організації та техніка проведення).
11. Використання засобів автоматизації модульного тестування.
12. Модульне, інтеграційне та системне тестування. Види та особливості проведення.
13. Документи, які супроводжують процес тестування – план тестування, тест-кейси, звіт про виявлені помилки. Призначення та структура документів.
14. Використання графів при тестуванні методом «чорного ящика». Структура графів і його основні властивості.
15. Особливості тестування об'єктно-орієнтованих програм.
16. Стандарти, що регламентують процес тестування. Коротка характеристика.
17. Граф потоку керування. Типовий алгоритм тестування графа.
18. Вимоги до ПЗ – визначення. Функціональні та нефункціональні вимоги.
19. Тестування, яке базується на структурі – різновиди покриття коду тестами та їх особливості.
20. Валідація та верифікація програмного забезпечення.
21. ISO/IEC/IEEE 29119 – стисла характеристика та основні положення.
22. ISO/IEC 25010 – стисла характеристика та основні положення.
23. Автоматизоване тестування у навантажувальному тестуванні.
24. Автоматизоване тестування у тестуванні БД.
25. Автоматизоване тестування у тестуванні web-застосунків.

26. Автоматизоване тестування у тестуванні мобільних застосунків.
27. Автоматизоване тестування у модульному тестуванні.
28. Автоматизоване тестування у інтеграційному тестуванні.
29. Тестування, яке базується на проєктних ризиках.
30. Базові принципи тестування.
31. Тестування та Quality Assurance.
32. Техніки тестування , які базуються на досвіді тестувальника.
33. План тестування (master test plan) – базові концепції, типова структура.
34. Типи дефектів.
35. Системи автоматизованого обліку дефектів.
36. Діаграми причинно-наслідкових зв'язків.
37. Статичне та динамічне тестування – порівняльний аналіз.
38. Регресійне тестування – базові положення, критерії для прийняття рішень по обсягу.
39. Тестування конфігурацій.
40. TDD та особливості його використання.
41. Типи покриття коду тестами.
42. Можливості JUnit Jupiter 5.4 для тестування.
43. Можливості TestNG для тестування.
44. Можливості Selenium Server для навантажувального та конфігураційного тестування.
45. Test Maturity Model Integrated (TMMi) – базові положення.
46. Використання прецедентів для дизайну тестів.
47. Data Driven Test – тести орієнтовані на дані.
48. Test Case – типова структура, критерії якості.
49. Доменна модель та модель вимог.
50. FURPS+ та інші моделі якості ПЗ.

5 НАВЧАЛЬНО-МЕТОДИЧНІ МАТЕРІАЛИ

До дисципліни «Об'єктно-орієнтоване програмування»

1. Albahari J. C# 9.0 in a Nutshell: The Definitive Reference. O'Reilly Media, 2021. 1060 p.
2. Stellman A., Greene J. Head First C#: A Learner's Guide to Real-World Programming with C# and .NET Core. 4th Edition. O'Reilly Media, 2021. 800 p.
3. Troelsen A., Japikse Ph. Pro C# 9 with .NET 5: Foundational Principles and Practices in Programming. 10th Edition. Apress, 2021. 1411 p.
4. Skeet J. C# in Depth. 4th Edition. Manning, 2019. 528 p.
5. Schildt H. Java: The Complete Reference. 12th Edition. McGraw Hill, 2021. 1280 p.
6. Loy M., Niemeyer P., Leuck D. Learning Java: An Introduction to Real-World Programming with Java. 5th Edition. O'Reilly Media, 2020. 518 p.
7. Evans B. Java in a Nutshell: A Desktop Quick Reference. 7th Edition. O'Reilly Media, 2019. 458 p.
8. Urma R. G., Fusco M., Mycroft A. Modern Java in Action: Lambdas, streams, functional and reactive programming. 2nd Edition. Manning, 2018. 592 p.
9. Liang Y. D. Introduction to Java Programming and Data Structures. 11th Edition. Pearson, 2019. 1232 p.

До дисципліни «Організація баз даних»

1. Пасічник В. В., Резниченко В. А. Організація баз даних та знань. К. : Видавн. група BHV, 2006. 384 с.
2. Кузнецов С. Д. Основы современных баз данных. [Електроний ресурс]. Режим доступу: URL: <https://www.citforum.ru/database/osbd/contents.shtml>. – Загол. з екрану.
3. Реляционная алгебра и реляционное исчисление. [Електроний ресурс]. Режим доступу: URL: <https://studfiles.net/preview/2928688/page:2/>
4. Jay Kreibich. Using SQLite. Small. Fast. Reliable. Choose Any Three. O'Reilly Media, 2010. 530 p.
5. Sibsankar Halder. SQLite Database System Design and Implementation (Second Edition, Version 2). O'Reilly, 2016. 264 p.
6. What Is SQLite? [Електроний ресурс]. Режим доступу: URL: <https://www.sqlite.org/index.html>. – Загол. з екрану.
7. Gorka Sadowksi, Philip Rathle. Fraud Detection: Discovering Connections Using Graph Databases: [Електронний документ]. – (<http://neo4j.com/blog/insurance-frauddetection-graph-database/>).

До дисципліни «Алгоритми та структури даних»

1. Ільман В.М., Іванов О.П., Панік Л.О. Алгоритми, дані і структури : навч. посіб. Дніпро : Дніпропет. нац. ун-т залізн. трансп. ім. акад. В. Лазаряна,

2019. 134 с.

2. Кірей К. О. Алгоритми та структури даних: конспект лекцій для студентів денної форми навчання спеціальності 121 «Інженерія програмного забезпечення». Миколаїв: ЧНУ ім. Петра Могили, 2021. 75 с.

3. Кірей К. О. Алгоритми та структури даних: методичні рекомендації для виконання лабораторних робіт студентами денної форми навчання спеціальності 121 «Інженерія програмного забезпечення». Миколаїв: ЧНУ ім. Петра Могили, 2019. 90 с.

4. Таненбаум Э., Бос Х. Современные операционные системы / пер. с англ. Леонтьева А., Малышева М., Вильчинский Н. Санкт-Петербург, Питер, 2015. 1120 с.

5. Уайс М. А. Организация структур данных и решение задач на C++ / пер. с англ. Москва : ЭКОМ Паблишес, 2008. 896 с.

6. Шаховська Н. Б., Голощук Р. О. Алгоритми і структури даних : навч. посіб. заг. ред. В. В. Пасічника. Львів : Магнолія 2006, 2014. 215 с.

7. Шилдт Г. C++: руководство для начинающих / пер. с англ. Москва, Вильямс, 2005. 672 с.

8. Stroustrup Bjarne. The C++ Programming Language (4th Edition). Addison-Wesley, 2013. 1368 p.

До дисципліни «Web development»

1. Будилов В. А. Основы программирования для Интернета. СПб. : БХВ-Петербург, 2003. 736 с.

2. Глушков С. В., Жакин И. А., Хачиров Т. С. Программирование Web-страниц. Х. : Фолио, 2005. 390 с.

3. Дэвид Скляр. Изучаем PHP 7: руководство по созданию интерактивных веб-сайтов. Диалектика-Вильямс, 2017. 464 с.

4. Robin Nixon. Learning PHP, MySQL & JavaScript, 4th Edition. With jQuery, CSS & HTML5. O'Reilly Media, 2014. 810 с.

5. Завдстра, Мэтт. PHP: Объекты, шаблоны и методики программирования, 4-е изд. Пер. с англ. М.: ООО "И.Д. Вильямс", 2015.

6. Максим Кузнецов, Игорь Симдянов. MySQL 5. СПб. : БХВ-Петербург, 2010. 342 с.

7. Дронов В. А. Laravel. Быстрая разработка современных динамических Web-сайтов на PHP, MySQL, HTML и CSS. М. : БХВ-Петербург, 2018. 160 с.

До дисципліни «Операційні системи»

1. Бондаренко М. Ф., Качко О. Г. Операційні системи : навч. Посібник. Х. : Компанія СМІТ, 2008. 432 с.

2. Шеховцев В. А. Операційні системи. К.: Видавнича група BHV, 2005. 576 с.

3. Шоттс У. Командная строка Linux. Полное руководство. СПб. : Питер, 2017. 480 с.: ил.

4. Блум Р., Бреснахэн К. Командная строка Linux и сценарии оболочки. Библия пользователя, 2-е изд.: Пер. с англ. М. : ООО «И.Д. Вильямс», 2012. 784 с.: ил.

5. Курячий Г. В., Маслинский К. А. Операционная система Linux: курс лекций. Учебное пособие. М. : ALT Linux; Издательство ДМК Пресс, 2010. 348 с.: ил.: 2-е изд., исправленное.

6. Немет Э., Снайдер Г., Хейн Т. Руководство администратора Linux, 2-е издание: Пер. с англ. М. : ООО «И.Д. Вильямс», 2011. 1072 с.: ил.

До дисципліни «Якість програмного забезпечення та тестування»

1. Myers, G. (2011) The Art of Software Testing, (3e), John Wiley & Sons: New York NY

2. Р. Савин (2007) Тестирование dot.com

3. Crispin, L. and Gregory, J. (2008) Agile Testing, Pearson Education: Boston MA

4. Jorgensen, P. (2014) Software Testing, A Craftsman's Approach (4e), CRC Press: Boca Raton FL

5. Авраменко А. С., Авраменко В. С., Косенюк Г. В. (2017) Тестування програмного забезпечення.

6. Kaner, C., Padmanabhan, S. and Hoffman, D. (2013) The Domain Testing Workbook, Context-Driven Press: New York NY

7. Craig, R. and Jaskiel, S. (2002) Systematic Software Testing, Artech House: Norwood MA

8. Black, R. (2017) Agile Testing Foundations, BCS Learning & Development Ltd: Swindon UK

Програма розглянута та затверджена на засіданні приймальної комісії університету (протокол № 6 від «03» квітня 2023 року).

Голова екзаменаційної
комісії



Альона ШВЕД

Відповідальний секретар
приймальної комісії



Вікторія ЧОРНА