ЗАТВЕРДЖУЮ
Голова приймальної комісії НТУ «ДП»,
Ректор _____ Олександр АЗЮКОВСЬКИЙ
« 02 » квітня 2026 р.

ПРОГРАМА
вступного екзамену зі спеціальності
F2 «Інженерія програмного забезпечення»
для вступу на навчання за ступенем доктора філософії

Уміння, що контролюються	Зміст програми
Будувати логічні висновки. Використовувати формальні мови і моделі алгоритмічних обчислень. Проектувати, розробляти та аналізувати алгоритми, оцінювати їх ефективність та складність, розв'язність та нерозв'язність алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем. Використовувати алгоритми програмні продукти, використовуючи методи класів та об'єктів.	1 **Теорія алгоритмів і математична логіка** 1.1 Алгоритми, вирази, обчислення в теорії алгоритмів 1.2 Одновимірні та багатовимірні масиви. Застосування при моделюванні прикладних задач 1.3 Алгоритми пошуку і сортування. Використання при розробці прикладних програм 1.4 Функції та процедури. Моделювання блоків кодів при розв'язуванні прикладних задач
Створювати та розробляти структурно-алгоритмічне забезпечення зі залученням математичних основ інженерії програмного забезпечення. Обґрунтовувати вибір інтерполяційного полінома для дослідження та наближеного уявлення функцій. Висовувати та обґрунтовувати гіпотези щодо законів розподілу дискретних та неперервних випадкових величин. Використовувати математичний апарат розв'язування екстремальних задач. Створювати та розробляти структурно-алгоритмічне забезпечення задач лінійної, нелінійної оптимізації та теорії ігор. Використовувати та аналізувати числові характеристики випадкових величин.	**2 Математичні основи інженерії програмного забезпечення** 2.1 Інтерполяція і наближення функцій. Інтерполяційний поліном Лагранжа 2.2 Безперервні випадкові величини. Функція щільності ймовірності та її властивості. Ймовірність влучення на заданий відрізок 2.3 Дискретні випадкові величини. Ряд розподілу дискретної СВ. Функція розподілу, її властивості і графік. Ймовірність попадання на напівінтервал 2.4 Нормальний розподіл. Крива Гауса. Функція розподілу нормального закону. Правило «трьох сігм»
Використовувати на практиці сучасні методи основ надійності програмних систем. Обґрунтовувати застосування критеріїв надійності програмних систем та характеристики їх відмов. Обґрунтовувати застосування основних характери стик безвідмовної роботи програмних засобів. Обґрунтовувати та застосовувати найбільш	**3 Основи теорії надійності програмних систем** 3.1 Фундаментальні поняття і визначення теорії надійності 3.2 Класифікація і характеристики відмов. Критерії надійності 3.3 Імовірність безвідмовної роботи. Густина розподілу часу безвідмовної роботи. Інтенсивність відмов. Середній час безвідмовної ро-

Уміння, що контролюються	Зміст програми
поширені закони розподілу часу до відмови. Використовувати та аналізувати математичні моделі функціонування технічних елементів і систем в сенсі їх надійності. Використовувати в практиці дослідження надійності програмних систем методи аналізу надійності технічних систем.	боти. Середній час роботи між відмовами та середній час відновлення 3.4 Найбільш поширені закони розподілу часу до відмови. Проблеми аналізу надійності складних технічних систем. Розробка моделей функціонування складної системи. Проблеми створення високонадійних систем
Використовувати на практиці основні структури програм на мовах програмування Python, C/C++ або Java. Вміти застосовувати основні алгоритмічні структури мов програмування. Вміти використовувати основні принципи об'єктно-орієнтованого програмування. Використовувати поняття масиву та його властивості, їх оголошення. Застосовувати при моделюванні прикладних задач одновимірні та багатовимірні масиви. Застосовувати оголошення класів мовою програмування Python, C++ або Java. Розробляти ієрархії класів з використанням інструментарію об'єктно-орієнтованого програмування (інкапсуляція, наслідування, поліморфізм).	**4 Мови та технології програмування** 4.1 Структура програм на мовах програмування Python, C/C++ або Java. Прості та складені оператори, приклади використання операторів. Поняття константи та змінної в мовах програмування, принципи виділення пам'яті для розміщення констант та змінних 4.2 Логічні операції та умовні оператори. Алгоритмічна структура розгалуження. Вибір із двох альтернатив. Блок схеми структур розгалуження. Вкладеність конструкцій вибору 4.3 Алгоритмічні конструкції повторення. Типи циклів. Блок схеми циклів. Організація циклів 4.4 Поняття функції (підпрограми) в мовах програмування. Оголошення та виклик функцій. Рекурсія функцій. Стандартні процедури та функції
Аналізувати великі дані та моделювати високорівневі абстракції у великих наборах даних різної природи. Проектувати сховища великих даних для видобутку даних і знань. Візуалізувати великі дані. Будувати і оцінювати регресивні моделі, що генеруються на основі великих даних. Знати і застосовувати на практиці методи і засоби управління знаннями. Знати та використовувати особливості кластерного аналізу та кластеризацію. Використовувати при обробці даних та застосовувати та практиці візуальний аналіз даних – Visual Mining, інтелектуальний аналіз текстів – Text Mining.	**5 Інтелектуальний аналіз даних** 5.1 Особливості застосування технології Data Mining. Поняття про системи підтримки прийняття рішень 5.2 Практичне застосування методів і засобів Data Mining. Моделі і методи Data Mining 5.3 Постановка завдання класифікації та регресії. Подання результатів класифікації та регресії. Методи побудови правил класифікації, дерев рішень, математичних функцій 5.4 Прогнозування часових рядів. Кладистика – спеціальний підхід до біологічної класифікації. Особливості пошуку асоціативних правил

Рекомендована література

1. Яковина В.С., Сенів М.М. Основи теорії надійності програмних систем. Львів: Видавництво Львівської політехніки, 2020. – 248 с.
2. Провотар О.І. Конкретна алгоритміка. – К. Наукова думка, 2017. – 162 с.
3. Настенко, Д. В. Об'єктно-орієнтоване програмування. Частина 1. Основи об'єктно-орієнтованого програмування на мові C# / Д. В. Настенко, А. Б. Нестерко. – Київ : НТУУ «КПІ», 2016. – 76 с.

4. Дослідження операцій. Конспект лекцій / Уклад.: О.І. Лисенко, І.В. Алєксєєва, – К: НТУУ «КПІ», 2016. – 196 с
5. Грицюк Ю.І. Аналіз вимог до програмного забезпечення. Львів: Видавництво Львівської політехніки, 2018. – 456 с.
6. Мельник Р.А. Програмування веб-застосувань. Львів: Видавництво Львівської політехніки, 2018. – 248 с.
7. Сенів М.М. Безпека програм та даних. – Львів: Видавництво Львівської політехніки, 2015. – 256 с.
8. Яковина В. С. Моделі, методи та засоби аналізу надійності програмних систем : монографія / Яковина В. С., Федасюк Д. В., Сенів М. М., Нитребич О. О. – Львів : Видавництво Львівської політехніки, 2015. – 220 с.
9. Яковина В.С. Методи аналізу надійності програмного забезпечення з урахуванням їх складності. Радіоелектронні і комп'ютерні системи. – 2015. – № 2(72). – С. 252 – 255.
10. Грицюк Ю.І. Інтелектуальний аналіз даних і процесів : навч. посібник / Ю.І.Грицюк. – Львів : Вид-во НУ Львівська політехніка;, 2018. – 440 с.
11. Костюченко А.О. Основи програмування мовою Python: навчальний посібник. Ч.: ФОП Баликіна С.М., 2020. –180 с.
12. Креневич А.П. Python у прикладах і задачах. Частина 1. Структурне програмування Навчальний посібник із дисципліни Інформатика та програмування - К. ВПЦ Київський Університет, 2017. - 206 с
13. Лавріщева К.М., Нікітченко М.С., Омельчук Л.Л.. Технологія програмування інформаційних систем. Підручник (гриф МОН України). – Киев: ВПЦ "Київський університет", 2015. – 367 с.
14. By Anu G. Aggarwal , Abhishek Tandon , Hoang Pham, Optimization Models in Software Reliability/ Springer, Berlin , 2022. – 373 p.
15. Бублик В.В. Об'єктно-орієнтоване програмування Підручник. / К.: ІТкнига, 2015. — 624 с.
16. Марченко О.О. Актуальні проблеми Data Mining: Навчальний посібник для студентів факультету комп'ютерних наук та кібернетики. / Марченко О.О., Россада Т.В. /Київ. -2017. -150 с.

Критерії оцінювання окремих завдань білета

Кожне теоретичне тестове завдання білета оцінюється 1 або 2 балами, а практичне та завдання на відповідність – 5 балами, виходячи з критеріїв:

а) однобальний теоретичний тест:

0 – вибір варіанта відповіді помилковий або обрано більш одного варіанта відповіді;

1 – обраний правильний варіант відповіді.

б) двобальний теоретичний тест:

0 – вибір варіантів відповідей помилковий або обрано більш трьох варіантів;

1 – лише один правильний варіант відповіді з двох обраних або два з трьох обраних;

2 – обрані тільки правильні варіанти відповідей.

в) практичне розрахункове завдання (задача):

0 – задача не вирішувалася, або були використані формули з грубими помилками, або як такі, що не належать до суті задачі;

1 – задача вирішувалася, але в підсумку були приведені тільки загальні формули та міркування або допущені грубі помилки у використанні формул;

2 – задача вирішувалася, але допущена груба помилка у формулі або в її використанні;

3 – задача вирішена в загальному виді, або містить грубу помилку в розрахунках, або ж відсутня пряма відповідь на запитання;

4 – задача вирішена в цілому правильно, але без відповідних пояснень, або допущена незначна помилка (неточність);

5 – задача вирішена правильно з відповідними поясненнями.

г) практичне завдання на програмування:

0 – програма відсутня, або у програмі використаний неправильний алгоритм, або використані дані, яких немає в умові задачі;

1 – у програмі конструкції мови або службові слова використано синтаксично неправильно, або програму неможливо змістовно інтерпретувати;

2 – у програмі деякі конструкції мови або службові слова використано синтаксично неправильно, в цілому програму можливо змістовно інтерпретувати;

3 – у програмі всі конструкції мови або службові слова використані синтаксично правильно, але допущені деякі помилки (відсутність необхідної кількості дужок або крапок з комою тощо);

4 – програма виконана повністю правильно, але відсутні коментарі;

5 – програма виконана повністю правильно і з відповідними коментарями.

Структура білета

Білет містить 30 однобальних теоретичних тестів, 5 двобальних та 12 п'ятибальних практичних розрахункових завдань, які охоплюють всі змістовні модулі програми іспиту. У підсумку максимальна сума балів білета складає 100 балів: 40 – за теоретичну частину та 60 – за практичну.

Шкала оцінювання білета

Вступний екзамен оцінюється за шкалою 100-200 балів. Мінімальний позитивний результат іспиту за виконання завдань білета (кваліфікаційний мінімум) складає 25 балів. Ця кількість балів відповідає екзаменаційній оцінки 100 шкали оцінювання. Переведення балів за виконання завдань білета вступного випробування до шкали 100-200 виконується відповідно до таблиці 5.15 додатка 5 Правил прийому до НТУ «Дніпровська політехніка». Вступники, які за результатами іспиту набрали менш ніж кваліфікаційний мінімум, позбавляються права участі в конкурсі.

Приклади екзаменаційних завдань білета

а) однобальний теоретичний тест:

Розбиття задачі на окремі кроки є виконанням властивості алгоритму:

а) зрозумілості, б) масовості,

в) дискретності, г) однозначності.

б) двобальний теоретичний тест:

За допомогою нормального розподілу можна описати ймовірність відмови об'єкта внаслідок його:

а) старіння, б) нормальної експлуатації, в) приробітку,
г) зносу, д) експериментальних випробувань, е) передачі замовнику.

в) практичне розрахункове завдання (задача):

Час безвідмовної роботи програмного забезпечення підкоряється закону Вейбулла з параметрами $\alpha = 1{,}5$; $\lambda_0 = 10^{-4}\frac{1}{год}$, час роботи t = 100 годин. Визначити ймовірність безвідмовної роботи P(t), частоту відмов f(t), інтенсивність відмов $\lambda(t)$, середній час безвідмовної роботи T_1 та статистичні характеристики закону розподілу.

г) практичне завдання на програмування:

Скласти код мовою Python для визначення метрику відстані Евкліда D – відстань між двома екземплярами у двомірному просторі, якщо A та B – екземпляри даних, x та y – атрибути екземплярів. При цьому: $A_x=2$, $A_y=3$, $B_x=4$, $B_y=5$.