

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХЕРСОНСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ

ЗАТВЕРДЖУЮ

Голова Приймальної комісії

ректор Херсонського державного університету

Александр СПИВАКОВСЬКИЙ

«05» червня 2026 р.



ПРОГРАМА

додакового вступного випробування зі спеціальності
для здобуття ступеня доктора філософії (PhD)
на основі повної вищої освіти (магістра)
(денна,вечірня, заочна форми навчання)

Галузь знань: F Інформаційні технології

Наукова спеціальність: F2 Інженерія програмного забезпечення

Освітньо-наукова програма: Інженерія програмного забезпечення

Затверджено на засіданні кафедри комп'ютерних наук та програмної інженерії
(протокол № 8 від 17 квітня 2026 року)

Укладач програми



Володимир ПЕСЧАНЕНКО

1. Загальні положення

Програма додаткового вступного екзамену зі спеціальності F2 Інженерія програмного забезпечення, для абітурієнтів які вступають на для здобуття ступеня доктора філософії (PhD) на основі повної вищої освіти (с.в.о. «магістр») здобутої за іншою спеціальністю розроблена відповідно до освітньонаукової програми третього освітньо-наукового рівня підготовки докторів філософії зі спеціальності F2 Інженерія програмного забезпечення.

Організація та проведення фахових вступних випробувань відбувається у порядку визначеному у Положенні про приймальну комісію Херсонського державного університету.

Мета вступного випробування – відбір претендентів на навчання за Зим науково-освітнім рівнем «доктор філософії».

Форма вступного випробування - вступне випробування проводиться письмово.

Тривалість вступного випробування – на виконання відведено 180 хвилин.

Результат вступного випробування оцінюється за шкалою від 100 до 200 балів.

Перепусткою на письмову роботу є Аркуш результатів вступних випробувань, паспорт.

Під час проведення вступного випробування не допускається користування електронними приладами, підручниками, навчальними посібниками та іншими матеріалами, якщо це не передбачено рішенням Приймальної комісії. У разі використання вступником під час вступного випробування сторонніх джерел інформації (у тому числі підказки) він

відсторонюється від участі у випробуваннях, про що складається акт. На екзаменаційній роботі такого вступника член фахової атестаційної комісії вказує причину відсторонення та час. При перевірці така робота дешифрується і за неї виставляється оцінка менше мінімальної кількості балів, визначеної Приймальною комісією та Правилами прийому, для допуску до участі в конкурсі або зарахування на навчання поза конкурсом, незважаючи на обсяг і зміст написаного.

Вступники, які не з'явились на вступне випробування без поважних причин у зазначений за розкладом час, до участі у подальших іспитах і конкурсі не допускаються.

ЗМІСТ ПРОГРАМИ

1. Загальні положення	3
2. Зміст програми.....	5
3. Перелік питань, що виносяться на додаткове вступне випробування	6
4. Список рекомендованої літератури	13
5. Критерії оцінювання знань додаткового вступного випробування	15

2. Перелік питань, що виносяться на додаткове вступне випробування

Структура мови програмування. (МП).

1. Синтаксис і семантика описів алгоритмів. Формальні засоби описів синтаксису МП. Функціональна і операційна семантика МП.
2. Обчислювальні структури (багатосортні алгебри) як формальні засоби опису даних. Носії та сигнатури, форми запису. Терми. Інтерпретація як семантика обчислювальних структур.
3. Оператори управління МП. Абстрактні типи даних та структури даних.
4. Методологія процедурного програмування.
Структуризація алгоритму в термінах процедур і функцій. Синтаксис описів і семантика виконання.
5. Локалізація даних. Техніка програмування в термінах процедур і функцій.
6. Рекурсивні описи процедур і функцій. Індуктивне тлумачення рекурсивних описів функцій.
7. Поняття доведення програм Передумови, постумови і інваріанти циклів.

Алгоритми і структури даних

1. Основні абстрактні типи даних АТД Список. Реалізація списків. АТД
Стек. Реалізація стеків. АТД Черга. Реалізація черг. АТД Відображення.

Реалізація відображень.

2. Означення дерева. Різні стратегії обходу дерев. АТД
Дерево. Реалізація дерев. Представлення дерев за допомогою масивів.
Реалізація дерев за допомогою зв'язаних списків.

3. Навантажені дерева. збалансованими деревами. 2-3 дерева
Реалізація множин збалансованими деревами. Алгоритм Зіва-Лемпела
архівації.

4. Орієнтовані графи. Різні методи представлення орграфа в
структурах даних. Обходи орграфа. Пошук в глибину та у ширину.

5. Задача пошуку найкоротшого шляху. Алгоритм
Дейкстри.

Обґрунтування та аналіз алгоритму Дейкстри.

6. Пошук найкоротших шляхів між усіма парами вершин
графа..

Алгоритм Флойда. Порівняння алгоритмів Флойда та Дейкстри.

7. Основні оператори множин Введення в множини.
Оператори АТД,
засновані на множинах. Реалізація множин за допомогою бінарних векторів.

8. Реалізація множин за допомогою зв'язаних списків.
Словники.

Реалізація словників.

9. Хешування. Структури даних, засновані на хеш-
теблицях. Відкрите хешування. Закрите хешування. Оцінка
ефективності хеш-функцій.

10. Неорієнтовані графи. Основні означення. Представлення неорієнтованих графів. Алгоритм Прима. Алгоритм Краскала. Обходи неорієнтованих графів: пошук в глибину та ширину.
11. Сортування. Модель внутрішнього сортування. Прості схеми сортування. Швидке сортування. Пірамідальне сортування. Час виконання сортувань порівняннями. Порядкові статистики.
12. Задачі обчислювальної геометрії. Задача побудови опуклої лінійної оболонки точкової множини на площині. Прості алгоритми. Ефективні алгоритми.
13. Методи розробки алгоритмів. Метод «Розподіляй і володій». Складення графіка проведення тенісного турніру. Баланс підзадач.
14. Динамічне програмування. Ймовірність перемоги в спортивних турнірах. Задача триангуляції. Пошук рішень на основі таблиці.
15. Жадібні алгоритми. Алгоритм Хафмена оптимального кодування.
- Жадібні алгоритми як евристики.

Програмування (C++)

1. C++. Структура програми. Стандартні типи даних. Ініціалізація. Розмір даних і діапазон значень. Спеціальні символи. Константи. Оператори `enum` і `typedef`.

2. C++. Оператори. Блоки і складені оператори. Операція

привласнення. Математичні операції і вирази. Поєднання привласнення і математичних операцій. Інкремент і декремент. Пріоритет.

3. C++. Логічний тип в С і C++. Операції відношення. Логічні

оператори I, АБО, НЕ (&&, ||, !). Програмування розгалужень. Оператор if.

Потрійна операція. Оператор вибору варіантів switch.

4. C++. Програмування циклів. Цикли з передумовою і постумовою (while, do-while). Цикли з параметрами (for). Безконечні цикли.

5. C++. Функції. Повертання значення. Передача параметрів.

Локальні і глобальні змінні. Рекурсія. Параметри за умовчанням.

Перевантаження.

Inline-функції.

6. C++. Використання бібліотек функцій. Математичні функції. Функції бібліотеки stdlib. Генерація випадкових чисел. Функції для аналізу символів.

7. C++. Класи і об'єкти. Оголошення класу. Реалізація методів класу. Створення об'єктів. Конструктори і деструктори. Обґрунтування необхідності в приватних членах-даних. Методи доступу типа get і set. Функції-члени const.

8. C++. Адреси змінних. Використання операції &. Показчики. NULL-показчик. Операція *. Показчики void*. Блукаючі показчики або такі, що висять. Показчик this.

9. C++. Стік і вільна пам'ять. Оператори new і delete.
Перевірка

виділення пам'яті. Оператор ->. Константні покажчики і об'єкти. Витік пам'яті.
Класи з динамічним виділенням пам'яті.

10. C++. Посилання. Передача аргументів
функціям за

посиланням з використанням покажчиків і посилань. Передача за посиланням
і ефективність. Конструктор копіювання.

11. C++. Ініціалізація в конструкторах. Значення за
умовчанням у методах класу і конструкторах. Перевантаження методів
класу і конструкторів. Перевантаження математичних унарних і бінарних
операцій.

12. C++. Перевантаження операторів порівняння.
Перевантаження оператора привласнення та розробка
конструктора копіювання для класів з динамічним виділенням пам'яті.
Поверхнєве і глибоке копіювання.

13. C++. Масиви. Помилка індексації. Покажчики і масиви.
Адресна арифметика. Обробка масивів з використанням індексів і покажчиків.
Масиви у якості параметрів функцій.

14. C++. Масиви об'єктів. Масиви покажчиків на об'єкти.
Масиви
об'єктів у вільній пам'яті. Класи для захищених масивів.

Перевантаження оператора [].

15. C++. Масиви символів (рядки С-типа). Строкові класи.
Переваги використання строкового класу в порівнянні з рядками символів.
Використання стандартного класу string.

16. C++. Спадкування. Базові і похідні класи. Перевантаження

конструкторів в похідних класах. Перекривання методів. Прямий виклик методів базового класу. Віртуальні методи. Віртуальні деструктори.

Віртуальний конструктор копіювання. Раннє та пізнє зв'язування.

17. C++. Множинне спадкування. Частини об'єкту при множинному спадкуванні. Конструктори і деструктори при множинному спадкуванні.

Зняття неоднозначності. Віртуальне спадкування. Абстрактні типи даних.

18. C++. Статичні члени-дані і методи. Доступ до статичних членів класу. Підрахунок кількості об'єктів за допомогою статичних даних.

19. C++. Показчики на функції. Масиви показчиків на функції. Передача показчиків на функції іншим функціям. Використання typedef з показчиками на функції. Показчики на функції і класи.

20. C++. Агрегація. Делегування. Реалізація в термінах іншого. Приватне спадкування. Дружні класи. Дружні функції. Попереднє оголошення класу.

21. C++. Потоки введення/виводу. Введення з використанням cin. Перевантаження оператора << для класів. Вивід з використанням cout. Маніпулятори, прапори і інструкції форматування. Перевантаження оператора >> для класів.

22. C++. Об'єкти типа ofstream і ifstream. Відкриття і закриття текстових потоків. Перевірка стану. Зміна поведінки ofstream при відкритті файлів. Запис і читання даних.

23. C++. Двійкові файли. Запис і читання об'єктів. Здобуття

інформації з командного рядка запуску програми. Внутрішнє уведення/виведення. Об'єкти типу `stringstream` і `istringstream`.

24. C++. Препроцесор. Використання `#define` для об'яви імен. Захист заголовних файлів від повторного включення. Умовна компіляція і її використання. Макроси.

25. C++. Конфлікт імен. Файлові змінні. Оператори `extern` і `static`. Простір імен. Оператори `namespace` і `using`. Використання простору імен `std`. Заголовні файли нового типу.

26. C++. Шаблони. Визначення шаблону класу. Реалізація шаблону класу. Використання шаблону класу для зберігання даних різних типів. Шаблони функцій.

27. C++. Виключення і обробка помилок. Викид виключення за допомогою `throw`. Використання блоків `try` і `catch`. Множинні специфікації `catch`.

28. C++. Інформація про тип підчас виконання програми (RTTI).

Оператор `typeid`. Виключення `bad_typeid`. Програмування з використанням RTTI.

29. C++. Операції приведення типу. Оператор `dynamic_cast`. Виключення `bad_cast`. Оператори `const_cast`, `static_cast`, `reinterpret_cast`. Порівняння старої і нової форм перетворення типів.

30. C++. Маніпуляції з бітами. Побітові операції. Установки, очищення і тестування бітів. Класи з бітовими полями. Об'єднання (`union`).

Об'єктно-орієнтоване програмування.

1. Базові поняття ОО підходу. Поняття класу. Протокол класу. Об'єкт класу. Дані-члени, функції-члени.
2. Статичні члени класу. Приватні, захищені, відкриті члени класу.
3. Конструктор, деструктор, функції-селектори, модифікатори, помічники.
4. Спадкування, його види.
5. Перевизначення методів у похідних класах.
6. Поняття поліморфізму. Раннє та пізнє зв'язування.
7. Віртуальні функції. Чисті віртуальні функції. Поняття абстрактного класу. Віртуальні деструктори.
8. Множинне спадкування, його види та правила застосування. Друзі класу. Правила визначення та засоби використання. Подружні класи, та функції.
9. Перевантаження операторів.

4. Список рекомендованої літератури

Алгоритми і структури даних

1. Львов М.С., Співаковський О.В. Вступ до об'єктноорієнтованого програмування. – Херсон: ХДПУ. – 240 с.
2. Фаронов В.В. Delphi 6. Учебний курс. М.: Издатель Молгачева С.В., 672 с.
3. Т. Бадд. Объектноориентированное программирование в действии / пер. С англ. СПб.: Питер, 1997. – 464 с.
4. Г. Буч. Объектноориентированный анализ и проектирование с примерами приложений на C++, 2ое изд./пер. с англ. М.: «Издательство Бином», СПб.: «Невский диалект», 1998 г. – 560 с.

5. С. Шлеер, С. Мэллор. Объектноориентированный анализ: моделирование мира в состояниях. Киев: Диалектика, 1993. – 240 с.

Интернет ресурси

6. Алгоритми та структури даних.–[Електронний ресурс]. –Режим доступу: <http://ksuonline.kspu.edu/>
7. Алгоритми та структури даних– [Електронний ресурс]. –Режим доступу: <http://eadnurt.diit.edu.ua/>

Програмування (C++)

Основна література

1. Трофименко О.Г., Прокоп Ю.В., Швайко І.Г., Буката Л.М., Косирева Л.А., Леонов Ю. Г., Ясинський В. В.. «C++. Основи програмування. Теорія та практика: підручник.». - 2010.
2. Трофименко О.Г. C++. Алгоритмізація та програмування : підручник / О.Г. Трофименко, Ю.В. Прокоп, Н.І. Логінова, О.В. Задерейко. 2ге вид. перероб. і доповн. – Одеса : Фенікс, 2019. – 477 с.
3. Дейтел Х.М., Дейтел П.Дж. Как программировать на C++; пер с англ. М.: ООО "Бином-Пресс", 2008. 1456 с. 4. Интегрированное средовище розроблення Visual Studio. URL: <https://msdn.microsoft.com/ru-ru/library/dn762121.aspx>
5. Керівництво по .NET Framework. URL: <https://docs.microsoft.com/ruru/dotnet/framework/index>
6. Либерти Дж., Джонс Б. Освой самостоятельно C++ за 21 день. 5-е изд. М.: Вильямс, 2009. 784 с.
7. Липшман С.Б., Лажойе Ж. Язык программирования C++. Полное руководство. Пер. с англ. СПб.: Невский диалект, 2011. 1105 с.
8. Прата Ст. Язык программирования C++. Лекции и упражнения: учебник; пер. с англ. М.: Диалектика-Вильямс, 2015. 928 с.
9. Страуструп Б. Язык программирования C++. Специальное издание; пер. с англ. М.: ООО "Бином-Пресс", 2005. 1104 с.

10. Страуструп Б. Дизайн и эволюция C++; пер. с англ. М. : ДМК Пресс; СПб.: Питер, 2007. 445 с. 11. Тип системы C++ (современный C++) // Microsoft Developer Network. URL: <https://msdn.microsoft.com/ruru/library/hh279663.aspx>
12. Довгалець С. М. Алгоритмічні мови та програмування. Частина 1. Основи інформатики та комп'ютерної техніки. Навчальний посібник / С. М. Довгалець, Р. В. Маслій. – Вінниця : ВНТУ, 2009. – 116 с.

Додаткова література

13. Кнут Д. Искусство программирования, том 1. Основные алгоритмы – The Art of Computer Programming, vol.1. Fundamental Algorithms / Д. Кнут. – М. : "Вильямс", 2006. – 720 с.
14. Яковенко А. Основи програмування: методичні вказівки до виконання комп'ютерних практикумів з дисципліни "Основи програмування". Основи програмування мовою Python / А. В. Яковенко. – Київ : НТУУ "КПІ ім. І. Сікорського", 2017. – 87 с.
15. Любанович Б. Python. Простой Python. Современный стиль программирования / Б. Любанович. – СПб. : Питер, 2016. – 480 с.
16. Федоров Д. Ю. Основы программирования на примере языка Python : учеб.пособие / Д. Ю. Федоров. – СПб. : Питер, 2016. – 176 с.

Интернет-ресурси

1. Основи програмування <http://www.ksuonline.ksu.ks.ua>
2. Python for Informatics <http://www.thinkpython.com>
3. Coursera <https://www.coursera.org/>
4. Codecademy <https://www.codecademy.com/>

Об'єктноорієнтоване програмування

1. Львов М.С., Співаковський О.В. Вступ до об'єктноорієнтованого програмування. Херсон: ХДПУ. 240 с.
2. Фаронов В.В. Delphi 6. Учебный курс. М.: Издатель Молгачева С.В., 672 с.

3. Т. Бадд. Объектноориентированное программирование в действии / пер. с англ. СПб.:Питер, 1997. 464 с.

4. Г. Буч. Объектноориентированный анализ и проектирование с примерами приложений на C++, 2ое изд./пер. с англ. М.: «Издательство Бином», СПб.: «Невский диалект», 1998 г. 560 с.

5. С. Шлеер, С. Мэллор. Объектноориентированный анализ: моделирование мира в состояниях. Киев: Диалектика, 1993. 240 с.

5. Критерії оцінювання знань додаткового вступного випробування

Оцінка за шкалою ECTS	
Оцінка	Пояснення
190-200	«Відмінно» – теоретичний зміст питання розкрито повністю, необхідні практичні навички роботи з навчальним матеріалом повністю сформовані, всі навчальні завдання, що передбачені робочою навчальною програмою, виконані в повному обсязі, відмінна відповідь без помилок або з однією незначною помилкою.
182-189	«Дуже добре» – теоретичний зміст питання розкрито повністю, необхідні практичні навички роботи з навчальним матеріалом в основному сформовані, всі навчальні завдання, що передбачені робочою навчальною програмою, виконані, якість виконання більшості з них оцінено числом балів, близьким до максимального, відповідь має дві-три незначні помилки.
174-181	«Добре» – теоретичний зміст питання розкрито повністю, практичні навички роботи з навчальним матеріалом в основному сформовані, всі навчальні завдання, що передбачені робочою навчальною програмою, виконані, якість виконання жодного з них не оцінено мінімальним числом балів, деякі види завдань виконані з помилками, відповідь має декілька незначних помилок або одну-дві значні помилки.
164-173	«Задовільно» – теоретичний зміст питання розкрито не повністю, але прогалини в знаннях не носять істотного (системного) характеру, необхідні практичні навички роботи з навчальним матеріалом в основному сформовані, більшість передбачених робочою навчальною програмою навчальних завдань виконана, деякі з виконаних завдань містять помилки, відповідь з трьома значними помилками.
160-163	«Достатньо» – теоретичний зміст питання розкрито частково, деякі практичні навички роботи з навчальним матеріалом не сформовані, частина передбачених робочою навчальною програмою завдань не виконана, або якість виконання деяких з них оцінено числом балів, близьким до мінімального, відповідь (в усній або письмовій формі) фрагментарна, непослідовна.

100-159	«Умовно незадовільно» – теоретичний зміст питання розкрито частково, необхідні практичні навички роботи з навчальним матеріалом не сформовані, більшість передбачених робочою навчальною програмою завдань не виконано або якість їх виконання оцінено числом балів, близьким до мінімального; при додатковій самостійній роботі над матеріалом дисципліни можливе підвищення якості виконання навчальних завдань (з можливістю повторного складання); робота, що потребує доопрацювання.
0-99	«Безумовно незадовільно» – теоретичний зміст питання не розкрито, необхідні практичні навички роботи з навчальним матеріалом не сформовані, всі виконані навчальні завдання містять грубі помилки, додаткова самостійна робота над матеріалом курсу не приведе до значимого підвищення якості виконання навчальних завдань; робота, що потребує повної переробки.