

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
БУДІВНИЦТВА І АРХІТЕКТУРИ

магістр

Факультет:  
автоматизації і інформаційних технологій

«Затверджую»  
Голова приймальної комісії  
Ректор  
Петро КУЛІКОВ



**ПРОГРАМА**  
вступних фахових випробувань  
до вступу на навчання для отримання  
ступеня магістра зі спеціальності  
**121. «ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ»**  
за освітньо-професійною програмою  
**«РОЗПОДІЛЕНІ ПРОГРАМНІ СИСТЕМИ І ТЕХНОЛОГІЇ»**  
галузі знань 12. «Інформаційні технології»

Затверджено на засіданні  
приймальної комісії, протокол  
№ 4 від « 24 » квітня 2023 р.

## ЗАГАЛЬНІ ПОЛОЖЕННЯ

Вступні випробування проводяться фаховою атестаційною комісією для осіб, які закінчили ЗВО та отримали диплом за освітнім ступенем «бакалавр» (освітньо-кваліфікаційним ступенем «спеціаліст», «магістр») і вступають на спеціальність 121. «Інженерія програмного забезпечення» відповідно правилам прийому до КНУБА на 2023 рік. Бажаючі навчатися для отримання ступеня магістра складають, як одну із умов правил прийому, контрольні заходи у формі тестування з фахових дисциплін «Об'єктно-орієнтоване програмування» і «Організація баз даних та знань».

Другий (магістерський) освітній рівень є освітньо-професійним і обов'язковим для продовження навчання за однією зі спеціальностей в аспірантурі.

Студенти спеціальності 121. «Інженерія програмного забезпечення» опановують знання, уміння та навички, які дозволяють їм: створення програмного забезпечення, у тому числі у сфері інтернет технологій, мережевого і комунікаційного програмного забезпечення; проектування інтерактивних інтерфейсів користувача; проектування і розробка баз даних; тестування та забезпечення якості програмного забезпечення; аналіз проектів та програм готових систем, аналіз потреб та проблем користувачів, пошук найоптимальніших рішень; розробка, виготовлення, постачання та документування програмного забезпечення на замовлення користувачів; надання консультацій у сфері проектування і розробки програмного забезпечення.

Місцем роботи випускників можуть бути організації, які займаються розробкою та супроводом програмного забезпечення, незалежні агенції з тестування програмного забезпечення.

Після закінчення навчання випускники отримують диплом та освітню кваліфікацію – магістр з інженерії програмного забезпечення.

## 2. ЗМІСТ ПРОГРАМИ ВСТУПНОГО ВИПРОБУВАННЯ

### 2.1. Дисципліна «Об'єктно-орієнтоване програмування»

- 2.1.1 Класифікація мов програмування.
- 2.1.2 Поняття об'єкту та його порівняння зі структурами даних та алгоритмів в мові С.
- 2.1.3 Основні переваги визначення методів у мові С#.
- 2.1.4 Характеристика каркасу Framework.Net .
- 2.1.5 Створення багаторівневої ієрархії класів у мові С#.
- 2.1.6 Об'єктно-орієнтований аналіз та проектування .
- 2.1.7 Система типів мови С#.
- 2.1.8 Оператори new та delete для керування пам'яттю.
- 2.1.9 Перевантаження операторів true false у мові С#
- 2.1.10 Перевантаження бінарних операторів у мові С#
- 2.1.11 Перевантаження унарних операторів у мові С#.
- 2.1.12 Основні поняття ООП. Класи та об'єкти.
- 2.1.13 Клас та екземпляр. Найпростіший синтаксис означення класу.
- 2.1.14 Специфікатори доступу public, protected та private.
- 2.1.15 Застосування абстрактних класів у мові С#.
- 2.1.16 Рядки у мові С# основні концепції роботи з ними.
- 2.1.17 Використання класу StringBuilder для побудови рядків у мові С#
- 2.1.18 Поняття інкапсуляції. Поняття про захист внутрішніх даних об'єкту.
- 2.1.19 Метод як канал доступу до внутрішніх даних. Модель об'єкта як чорна скринька.
- 2.1.20 Конструктори, деструктори, їх роль та призначення.
- 2.1.21 Динамічне виділення пам'яті для об'єкту: виклик конструктора з оператору new.
- 2.1.22 Конструктори по замовчуванню та з параметрами. Особливості ініціалізації членів-даних.
- 2.1.23 Наслідування. Механізм наслідування членів-даних та методів.
- 2.1.24 Доступ до членів класу та наслідування у мові С#
- 2.1.25 Відношення клас-підклас та його зв'язок з відношенням абстрактне-конкретне. Сумісність типів знизу вгору.
- 2.1.26 Особливості виклику конструктора базового класу з конструктору надкласу. Порядок виклику конструкторів та деструкторів для об'єктів похідних класів.
- 2.1.27 Ієрархія класів. Множинне наслідування та його проблеми.

- 2.1.28 Поліморфізм та віртуальні функції. Поняття оголошеного та фактичного типу.
- 2.1.29 Механізм виклику віртуальної функції.
- 2.1.30 Дружні функції, окремі класи та класи в цілому, обхід механізмів захисту членів класу.
- 2.1.31 Переваги та недоліки використання механізму дружності в програмах з об'єктно-орієнтованою композицією.
- 2.1.32 Перевантаження операторів функціями та методами.
- 2.1.33 Віртуальні методи у мові C#.
- 2.1.34 Класи потоків введення-виведення. Управління форматом, модифікатори.
- 2.1.35 Шаблони функцій та класів.
- 2.1.36 Організація захищеного доступу у мові C#.
- 2.1.37 Перевантаження логічних операторів у мові C#.
- 2.1.38 Обробка виняткових ситуацій. Поняття про виняткову ситуацію.
- 2.1.39 Оператори try, throw і catch.
- 2.1.40 Порядок збудження та перехоплення виняткової ситуації.
- 2.1.41 Класи колекцій стандартної бібліотеки: послідовні контейнери.
- 2.1.42 Класи колекцій стандартної бібліотеки: асоціативні контейнери.
- 2.1.43 Основи стандартної бібліотеки: типи ітераторів та операції над ними.
- 2.1.44 Основи стандартної бібліотеки: узагальненні алгоритми. Види узагальнених алгоритмів. Приклад.
- 2.1.45 Основи стандартної бібліотеки: поняття функтора. Адаптери функторів.
- 2.1.46 Графічна мова UML. Діаграма прецедентів.
- 2.1.47 Графічна мова UML. Діаграма класів.
- 2.1.48 Графічна мова UML. Діаграма використання.
- 2.1.49 Графічна мова UML. Діаграма послідовностей.
- 2.1.50 Графічна мова UML. Діаграма діяльності.

## **ЛІТЕРАТУРА ДЛЯ ПІДГОТОВКИ**

1. О.В. Галкін, Л.О. Катеринич, О.С. Шкільняк. Програмування на JAVA 8: навчальний посібник. – К.: Логос, 2017. – 185 с.
2. Бегун А.В. Об'єктно-орієнтоване програмування. Навчальний посібник для самостійного вивчення дисципліни – К.: КНЕУ, 2008 – 130с.

3. О. Петрик. Об'єктно-орієнтоване програмування в середовищі С++: Навчальний посібник. Лабораторний практикум – Тернопіль, видавництво ТНТУ імені Івана Пулюя, 2011. – 188 с.

4. Кравець П.О. Об'єктно-орієнтоване програмування. Львів: Видавництво Львівської політехніки, 2012. – 624 с.

5. Коноваленко І.В., Марущак П.О., Савків В.Б. Програмування мовою С# 7.0: навчальний посібник. Тернопіль: Тернопільський національний технічний університет імені Івана Пулюя, 2017. 300 с.

6. Дібрівний О. А., Гребенюк В. В. Вступ до об'єктно-орієнтованого програмування С#: Навчальний посібник / Київ: Державний університет телекомунікацій, 2018, - 190с.

7. Кузнєцов М. С. Об'єктно-орієнтоване програмування з використанням UML та мови С++. Дніпропетровськ: НМетАУ, 2003. – 90 с.

## **2.2. Дисципліна «Організація баз даних і знань».**

2.2.1 Основні визначення та поняття бази даних і знань.

2.2.2 Поняття системи управління базою даних.

2.2.3 Основні поняття та визначення реляційної бази даних.

2.2.4 Поняття об'єктно-орієнтованої бази даних.

2.2.5 Представлення концептуальної моделі бази даних.

2.2.6 Функції виконання фактографічної інформаційної системи.

2.2.7 Моделі даних, що реалізує сучасні СУБД.

2.2.8 Поняття та визначення незалежність даних.

2.2.9 Вигляд представлення запиту в реляційному обчисленні зі змінними кортежами.

2.2.10 Проблеми, для вирішення яких необхідно виконувати нормалізацію бази даних.

2.2.11 Визначення і поняття першої нормальної форми.

2.2.12 Визначення і поняття другої нормальної форми..

2.2.13 Визначення і поняття третьої нормальної форми..

2.2.14 Визначення поняття “цілісність бази даних”.

2.2.15 Визначення поняття «відношення».

2.2.16 Визначення поняття «кортеж».

2.2.17 Визначення поняття “домен”.

2.2.18 Поняття «ключ» в реляційній базі даних.

- 2.2.19 Визначення поняття «зовнішній ключ».
- 2.2.20 Визначення поняття «реплікація».
- 2.2.21 Визначення поняття «тригер».
- 2.2.22 Визначення поняття “транзакція”.
- 2.2.23 Використання метадані в базах даних.
- 2.2.24 Особливості архітектури файл-сервер.
- 2.2.25 Реалізація, що дозволяє зробити індекси.
- 2.2.26 Дії, що передбаченні для забезпечення безпеки даних в базі даних.
- 2.2.27 Функції виконання представлення (view).
- 2.2.28 Архітектура програмно-технічних засобів розподілених СУБД.
- 2.2.29 Функції виконання сервер застосувань.
- 2.2.30 Використання технології, що передбачає реплікацію.
- 2.2.31 Використання технології, що передбачає фрагментацію.
- 2.2.32 Рівні ієрархії пристроїв пам’яті бази даних.
- 2.2.33 Використання кеш-пам’ять для потреб.
- 2.2.34 Визначення та поняття дамп бази даних.
- 2.2.35 Застосування мето-поле в базах даних.
- 2.2.36 Застосування В-дерева в базах даних.
- 2.2.37 Визначення “ADO”.
- 2.2.38 Визначення “OLE DB”.
- 2.2.39 Представлення інтерфейсу ISAPI/NSAPI.
- 2.2.40 Визначення “QBE”.
- 2.2.41 Визначення типу мов програмування, що відносяться до мови SQL.
- 2.2.42 Вигляд, що має правильний синтаксис для виключення пустих значень з таблиці.
- 2.2.43 Команди, які включають мову запитів до бази даних.
- 2.2.44 Команди, які включають мову маніпулювання з даними DML.
- 2.2.45 Команди, які на мові SQL застосовуються для визначення обмежень полів в базі даних.
- 2.2.46 Визначення “ACID”.
- 2.2.47 Визначення і поняття інкрементного поля.
- 2.2.48 Представлення програми – клієнт.
- 2.2.49 Правила перетворення ER-діаграм, які перетворюються в логічні моделі реляційної бази даних.
- 2.2.50 Правильність складання SQL-запитів до бази даних.
- 2.2.51 Правильність застосування команд SQL по створенню бази даних.

- 2.2.52 Правильність приведення логічної схеми бази даних до третьої нормальної форми.
- 2.2.53 Представлення “CORBA”.
- 2.2.54 Класична архітектура бази даних, яка включає запропоновану ANSI.
- 2.2.55 Основні поняття та визначення сховище даних.
- 2.2.56 Рівні ізоляції транзакцій і в чому їх особливості.
- 2.2.57 Головні властивості розподіленої бази даних.
- 2.2.58 Моделі або модель, яка підтримує об’єктно-реляційну базу даних.
- 2.2.59 Поняття хешування і для чого воно використовується.
- 2.2.60 Визначення репозиторій.
- 2.2.61 Функції, що виконують серіалізацію транзакцій.
- 2.2.62 Функції, що виконують файловий сервер.
- 2.2.63 Поняття, визначення “BLOB”.
- 2.2.64 Журнал реєстрації транзакцій, принцип використання.

## ЛІТЕРАТУРА ДЛЯ ПІДГОТОВКИ

1. Тарасов О. В. Використання мови SQL для роботи з сучасними системами керування базами даних / О. В. Тарасов, В. В. Федько, М. Ю. Лосєв. – Х. : Вид. ХНЕУ, 2013. – 348 с.
2. Тарасов О. В. Проектування баз даних : навч. посіб. / О. В. Тарасов, В. В. Федько, М. Ю. Лосєв. – Х. : Вид. ХНЕУ, 2011. – 200 с.
3. Анісімов А.В., Кулябко П.П. Інформаційні системи та бази даних: Навчальний посібник для студентів факультету комп’ютерних наук та кібернетики. – Київ. – 2017. – 110 с.
4. Гайна Г.А. Основи проектування баз даних: Навчальний посібник / Г.А. Гайна. – К. : КНУБА, 2005. – 204 с.
5. Мулеса О.Ю. Інформаційні системи та реляційні бази даних. Навч.посібник. – Ужгород: ДВНЗ УжНУ, Електронне видання, 2018. – 118 с.
6. Ковальчук А.М. Принципи проектування баз даних: Навчальний посібник. / Ковальчук А.М., Левицький В.Г. та ін. – Ж.: ЖДТУ, 2009. – 123с.
7. Atkinson P., Vieira R. Beginning Microsoft SQL Server 2012 Programming. - Wiley, 2016, – 864с.

### 3. СТРУКТУРА ЕКЗАМЕНАЦІЙНОГО БІЛЕТА. КРИТЕРІЇ ОЦІНЮВАННЯ

Оцінювання знань вступників на вступному випробуванні здійснюється за шкалою від 60 до 100 балів по кожній із зазначених дисциплін. Вступне випробування включає тестові завдання з дисциплін, кожне з яких налічує 20 питань. Кожне питання має чотири відповіді позначені літерами (цифрами), одна з яких вірна. Вступник обирає правильну відповідь до тестового питання та позначає її відповідною літерою (цифрою) напроти номера питання у стовпчику «відповідь» бланка-відповіді. Якщо вступник вирішив виправити відповідь на питання, то має внести зміну літерою (цифрою) у стовпчик «виправлена відповідь». Прийнятим до оцінювання буде запис внесений у стовпчик «виправлена відповідь». Викреслювати відповіді (літери, цифри) не дозволяється. Кожна правильна відповідь на питання оцінюється в 1 (один) бал, неправильна відповідь – 0 (нуль) балів і за таблицею переведення визначається конкурсна оцінка з фахового випробування.

Таблиця переведення результатів випробування (тестування) з фахової дисципліни із 20-ти бальної у 100-бальну шкалу оцінювання знань

Кількість вірних відповідей на запитання тестового завдання	Кількість балів за шкалою ECTS
<b>0...5</b>	<b>0</b> (незадовільно – контрольний захід нескладений)
<b>6</b>	<b>60</b>
<b>7</b>	<b>61</b>
<b>8</b>	<b>63</b>
<b>9</b>	<b>65</b>
<b>10</b>	<b>68</b>
<b>11</b>	<b>71</b>
<b>12</b>	<b>74</b>
<b>13</b>	<b>77</b>
<b>14</b>	<b>80</b>
<b>15</b>	<b>83</b>
<b>16</b>	<b>86</b>
<b>17</b>	<b>89</b>
<b>18</b>	<b>92</b>
<b>19</b>	<b>96</b>
<b>20</b>	<b>100</b>

Якщо вступник не склав контрольний захід хоча б по одній з дисциплін, вважається що він не пройшов фахове випробування та втрачає право брати участь в конкурсі на навчання для отримання ступеня магістра.

За результатами вступного випробування визначається сумарна кількість балів з зазначених дисциплін, на підставі якої фахова атестаційна комісія вносить на розгляд приймальної комісії університету затвердження результатів фахового випробування.

Голова фахової атестаційної комісії



Ігор РУСАН