

Міністерство освіти і науки України  
Харківський національний університет імені В.Н. Каразіна  
Кафедра штучного інтелекту та програмного забезпечення

“ЗАТВЕРДЖУЮ”



Проректор  
з науково-педагогічної роботи

Олександр ГОЛОВКО

» \_\_\_\_\_ 2022 р.

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

**Крос-платформне програмування**

рівень вищої освіти	<u>перший (бакалаврський) рівень</u>
галузь знань	<u>12 Інформаційні технології</u>
спеціальність	<u>122 Комп'ютерні науки</u>
освітня програма	<u>Комп'ютерні науки</u>
вид дисципліни	<u>обов'язкова</u>
факультет	<u>комп'ютерних наук</u>

2

Програму рекомендовано до затвердження Вченою радою факультету комп'ютерних наук  
«29» серпня 2022 року, протокол № 14

**РОЗРОБНИКИ ПРОГРАМИ:**

кандидат фізико-математичних наук, доцент кафедри штучного інтелекту та програмного  
забезпечення **Споров Олександр Євгенович**

Програму схвалено на засіданні кафедри штучного інтелекту та програмного забезпечення  
Протокол від «29» серпня 2022 року № 1

Завідувач кафедри штучного інтелекту та програмного забезпечення

  
Володимир КУКЛІН

Програму погоджено з гарантом освітньої програми «Комп'ютерні науки»

Гарант освітньої програми «Комп'ютерні науки»

  
Сергій БОГУЧАРСЬКИЙ

Програму погоджено методичною комісією факультету комп'ютерних наук

Протокол від «29» серпня 2022 року № 1

Голова методичної комісії факультету комп'ютерних наук

  
Анатолій БЕРДНІКОВ

## ВСТУП

Програма навчальної дисципліни «Крос-платформне програмування» складена відповідно до освітньо-професійної програми підготовки першого (бакалаврського) рівня вищої освіти, спеціальності 122 «Комп'ютерні науки».

### 1. Опис навчальної дисципліни

1.1. Метою навчальної дисципліни є огляд фундаментальних засад компонентної та розподіленої організації програм, так і прийомів практичного створення компонентних розподілених програмних продуктів на широко розповсюдженій мові крос-платформного програмування *Java*.

1.2. Основними завданнями вивчення дисципліни є:

- ознайомлення з архітектурами та стандартами компонентних моделей;
- ознайомлення з принципами проектування та розробки віддалених компонентів;
- ознайомлення з основами взаємодії різних програмних платформ в розподілених корпоративних інформаційних системах;
- ознайомлення з комунікаційними засобами розподілених обчислень;
- отримання умінь розв'язувати проблеми масштабованості.

В ході вивчення дисципліни у студента повинні формуватися наступні компетентності.

#### *Інтегральна компетентність:*

Здатність розв'язувати складні спеціалізовані задачі та практичні проблеми у галузі комп'ютерних наук або у процесі навчання, що передбачає застосування теорій та методів інформаційних технологій і характеризується комплексністю та невизначеністю умов.

#### *Загальні компетентності (ЗК).*

- ЗК1. Здатність до абстрактного мислення, аналізу та синтезу.
- ЗК2. Здатність застосовувати знання у практичних ситуаціях.
- ЗК3. Знання та розуміння предметної області та професійної діяльності.
- ЗК4. Здатність спілкуватися державною мовою як усно, так і письмово.
- ЗК5. Здатність спілкуватися іноземною мовою.
- ЗК6. Здатність вчитися й оволодівати сучасними знаннями.
- ЗК7. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
- ЗК8. Здатність генерувати нові ідеї (креативність).
- ЗК11. Здатність приймати обґрунтовані рішення.
- ЗК12. Здатність оцінювати та забезпечувати якість виконуваних робіт.

#### *Спеціальні (фахові, предметні) компетентності (ФК)*

- ФК1. Здатність до математичного формулювання та досліджування неперервних та дискретних математичних моделей, обґрунтування вибору методів і підходів для розв'язування теоретичних і прикладних задач у галузі комп'ютерних наук, аналізу та інтерпретування.
- ФК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного

моделювання предметних областей і створення програмних та інформаційних систем.

- ФК4. Здатність використовувати сучасні методи математичного моделювання об'єктів, процесів і явищ, розробляти моделі й алгоритми чисельного розв'язування задач математичного моделювання, враховувати похибки наближеного чисельного розв'язування професійних задач.
- ФК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.
- ФК9. Здатність реалізувати багаторівневу обчислювальну модель на основі архітектури клієнт-сервер, включаючи бази даних, знань і сховища даних, виконувати розподілену обробку великих наборів даних на кластерах стандартних серверів для забезпечення обчислювальних потреб користувачів, у тому числі на хмарних сервісах.

1.3. Кількість кредитів – 4

1.4. Загальна кількість годин - 120

1.5. Характеристика навчальної дисципліни	
Нормативна / за вибором	
денна форма навчання	Заочна (дистанційна) форма навчання
Рік підготовки	
2-й	
Семестр	
4-й	
Лекції	
32 год.	_____ год.
Практичні, семінарські заняття	
_____ год.	_____ год.
Лабораторні заняття	
32 год.	_____ год.
Самостійна робота	
26 год.	_____ год.
Індивідуальні завдання	
30 год.	_____ год.

1.6. Заплановані результати навчання

**знати:**

- основні технології, що лежать в основі та забезпечують можливість організації компонентної та розподіленої архітектури комп'ютерних програм;
- основи технології компонентного програмування та технології створення розподілених програм різної архітектури.

**вміти:**

- проектувати та створювати компонентні та розподілені програмні комплекси за допомогою широко розповсюдженій мові крос-платформного програмування *Java*.

В ході вивчення дисципліни у студента повинні формуватися наступні компетентності:

В результаті вивчення дисципліни у студента повинні формуватися наступні програмні результати навчання (ПРН).

- **ПР1.** Застосовувати знання основних форм і законів абстрактне-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.
- **ПР5.** Проектувати, розробляти та аналізувати алгоритми розв'язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислюваних функцій.
- **ПР6.** Використовувати методи чисельного диференціювання та інтегрування функцій, розв'язання звичайних диференціальних та інтегральних рівнянь, особливостей чисельних методів та можливостей їх адаптації до інженерних задач, мати навички програмної реалізації чисельних методів.
- **ПР9.** Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв'язання задач в галузі комп'ютерних наук.
- **ПР10.** Використовувати інструментальні засоби розробки клієнт-серверних застосувань, проектувати концептуальні, логічні та фізичні моделі баз даних, розробляти та оптимізувати запити до них, створювати розподілені бази даних, сховища та вітрини даних, бази знань, у тому числі на хмарних сервісах, із застосуванням мов веб-програмування.
- **ПР17.** Виконувати паралельні та розподілені обчислення, застосовувати чисельні методи та алгоритми для паралельних структур, мови паралельного програмування при розробці та експлуатації паралельного та розподіленого програмного забезпечення.

## **2. Тематичний план навчальної дисципліни**

*Розділ 1. Самоаналіз.*

*Тема 1. Рефлексія.*

Поняття рефлексії. Клас Class. Аналіз властивостей класів. Аналіз конструкторів ласу. Аналіз методів класу. Аналіз полів класу.

*Тема 2. Рефлексивні «показчики» на методи. Класи — посередники.*

Об'єкт Method. Перелік аргументів методу. Рефлексивне створення нових об'єктів та масивів. Завантажувач класів. Поняття про Proxy-classes. Обробник викликів. Властивості та використання класів-посередників.

*Розділ 2. Персистентність.*

*Тема 1. Стандартний інтерфейс серіалізації Serializable.*

Поток введення-виведення. Об'єктні потоки. Стандартний інтерфейс серіалізації. Посилання на об'єкти, що не серіалізуються. Transient поля. Версії серіалізації.

*Тема 2. Розширений інтерфейс серіалізації Externalizable.*

Розробка формату серіалізації. Методи readObject та writeObject. Методи defaultReadObject та defaultWriteObject. Розширений інтерфейс серіалізації.

*Тема 3. Стандарт XML. Схема DTD та SXD. Зв'язок документа та схеми.*

Стандарт XML. Базові елементи XML документу. Коректні XML документи. Схеми

XML документів. Зв'язок документу та схеми.

*Тема 4. Обробка XML документу. SAX – парсери.*

Різні підходи до обробки XML документів. Simple API for XML. Основні властивості; переваги та недоліки. Базові повідомлення. Обробники повідомлень. Валідуючі та не валідуючі SAX – парсери.

*Тема 5. Модель DOM. DOM парсери.*

Специфікація Document Object Model. Стандарти DOM. Модель документу. Побудова дерева документу. Класи, методи та атрибути цих методів, що призначені для аналізу структури документів та роботи із представленням документів у вигляді дерева. Валідуючі та не валідуючі DOM – парсери. Основні переваги валідуючих DOM – парсерів. Збереження XML документів.

*Розділ 3. Компонентне та мережеве програмування.*

*Тема 1. Компонентний підхід до створення програмних продуктів.*

Основні технології програмування. Компонентно орієнтоване програмування як розвиток об'єктно орієнтованого програмування. Компонент. Властивості компонента. Визначення компонентів. Специфікація інтерфейсу як контракту. Основні вимоги до компонентного програмного забезпечення.

*Тема 2. Компонентна технологія Java Beans.*

Основні поняття компонентної ідеології Java Beans. Компоненти Java Beans. Контейнери Java Beans. Властивості компонентів: скалярні, індексовані та булеві. Пов'язані та обмежені властивості. Події компонентів. Персистентність компонентів. Особливості налаштування та серіалізації компонентів. Серіалізація в XML файли.

*Тема 3. Мережеве програмування. Сокети.*

Мережа Інтернет. Протокол. Стек протоколів TCP/IP. Класи для представлення адреси. Протокол TCP/IP. Сокети TCP/IP. Базові правила роботи з потоковими сокетами. Основні переваги та недоліки. Передача об'єктів за допомогою сокетів через мережу. TCP/IP сервери та клієнти. Однопоточний та багатопоточний сервер. Розробка власного протоколу віддаленого виклику методів.

*Тема 4. Мережеве програмування. Датаграми.*

Датаграми. Основні переваги та недоліки датаграм. Класи, що представляють датаграми. Передача об'єктів та масивів об'єктів за допомогою датаграм. Датаграм-сервер. Датаграм-клієнт.

*Тема 5. Мережеве програмування. Багатоадресна передача інформації.*

Особливості роботи з великою кількістю клієнтів. Можливість організації багатоадресної (multicast) передачі інформації. Класи, що надають можливість одночасної роботи з декількома клієнтами. Особливості багатоадресної (multicast) передачі інформації.

*Розділ 4. Розподілене програмування.*

*Тема 1. Розподілена архітектура компонентних систем. Технологія Java RMI.*

Основні ознаки розподіленої системи. Фактори розвитку розподілених систем. Переваги та недоліки розподілених систем. Основні вимоги до розподілених систем. Проміжне середовище. Технологія розподіленого програмування Java RMI. Віддалені об'єкти. Основні інтерфейси та класи для створення розподілених доданків за технологією Java RMI. Структура найпростішого розподіленого доданку.

## Тема 2. Розгортання розподілених доданків Java RMI.

Віддалений виклик методу. Віддалені інтерфейси. Віддалена передача параметрів: заглушки та скелетони. Використання динамічних гроху. Служба реєстру RMI. Створення / запуск служби реєстру. Реєстація віддалених об'єктів в службі реєстру. Динамічне завантаження класів. Розгортання віддалених доданків на декількох вузлах мережі.

## Тема 3. Бази даних, як невід'ємна складова розподілених систем.

Бази даних – важливий компонент розподілених систем. Дво- та трьох- ланкові архітектури. Сервери баз даних. Простий стандартних сервер Java DB (Derby). Створення простої бази даних. Основні засоби мови SQL.

## Тема 4. Основи JDBC.

Основні принципи роботи з базами даних в Java. Основні класи та інтерфейси для взаємодії з базами даних. Створення автономного та розподіленого доданків, що взаємодіють з сервером бази даних Java DB.

### 3. Структура навчальної дисципліни

Назви розділів і тем	Кількість годин					
	денна форма					
	усього	у тому числі				
л		п	лаб.	інд.	с. р.	
1	2	3	4	5	6	7
<b>Розділ 1. Самоаналіз</b>						
Тема 1. Рефлексія.	6	2		2		2
Тема 2. Рефлексивні «показчики» на методи. Класи-посередники.	6	2		2		2
<b>Разом за розділом 1</b>	<b>12</b>	<b>4</b>		<b>4</b>		<b>4</b>
<b>Розділ 2. Персистентність</b>						
Тема 1 Стандартний інтерфейс серіалізації Serializable.	5	2		2		1
Тема 2. Розширений інтерфейс серіалізації Externalizable.	5	2		2		1
Тема 3. Стандарт XML. Схема DTD та SXD. Зв'язок документа та схеми.	5	2		2		1
Тема 4. Обробка XML документа. SAX – парсери.	6	2		2		2
Тема 5. Модель DOM. DOM парсери.	6	2		2		2
<b>Разом за розділом 2</b>	<b>27</b>	<b>10</b>		<b>10</b>		<b>7</b>
<b>Розділ 3. Компонентне та мережеве програмування</b>						
Тема 1. Компонентний підхід до створення програмних продуктів.	3	2				1
Розрахунково-графічна робота №1	15				15	
Тема 2. Компонентна технологія <i>Java Beans</i> .	6	2		2		2
Тема 3. Мережеве програмування. <i>Сокети</i> .	6	2		2		2
Тема 4. Мережеве програмування. <i>Датаграми</i> .	5	2		2		1
Тема 5. Мережеве програмування. <i>Multicast sockets</i> .	5	2		2		1
Контрольна робота	2			2		
<b>Разом за розділом 3</b>	<b>42</b>	<b>10</b>		<b>10</b>	<b>15</b>	<b>7</b>
<b>Розділ 4. Розподілене програмування</b>						
Тема 1. Розподілена архітектура компонентних систем.	6	2		2		2
Тема 2. Розгортання розподілених доданків Java RMI.	6	2		2		2
Тема 3. Бази даних, як невід'ємна складова розподілених систем	6	2		2		2
Розрахунково-графічна робота №2	15				15	
Тема 4. Основи JDBC	6	2		2		2
<b>Разом за розділом 4</b>	<b>39</b>	<b>8</b>		<b>8</b>	<b>15</b>	<b>8</b>
<b>Усього годин</b>	<b>120</b>	<b>32</b>		<b>32</b>	<b>30</b>	<b>26</b>

#### 4. Теми практичних (лабораторних) занять

№ з/п	Назва теми	Кількість годин
<b>Розділ 1. Самоаналіз</b>		
1	Об'єктно-орієнтоване програмування: система роботи з функціями	2
2	Рефлексивний аналіз класів	2
3	Створення об'єктів та масиву об'єктів за допомогою рефлексії. Рефлексивні «показчики» на методи. Динамічні Proxy-класи	2
<b>Розділ 2. Персистентність</b>		
4	Серіалізація системи за допомогою інтерфейсу Serializable	2
5	Серіалізація системи за допомогою інтерфейсу Externalizable	2
6	Створення XML документів та схем, зв'язок документа та схеми	2
7	Обробка XML документів за допомогою SAX парсеру	2
8	Обробка XML документів за допомогою DOM парсеру. Збереження документу.	2
<b>Розділ 3. Компонентне та мережеве програмування</b>		
9	Створення візуальних компонентів для зберігання та для графічного відображення даних	2
10	Створення програми з візуальним інтерфейсом користувача з стандартних та власних компонентів	2
11	TCP сокети. Мережева клієнт-серверна програма обробки інформації	2
12	Датаграми. Мережева клієнт-серверна програма передачі інформації	2
13	Багатоадресні сокети. Мережева клієнт-серверна програма багатоадресної передачі інформації	2
<b>Розділ 4. Розподілене програмування</b>		
14	Проектування розподіленої обчислювальної системи. Розробка віддалених серверних та клієнтських інтерфейсів. Створення клієнтської частини RMI обчислювальної програми	2
15	Створення серверної частини RMI обчислювальної програми. Запуск розподіленої програми з можливістю динамічної загрузки файлів, що відсутні	2
16	Основи JDBC. Створення розподіленої системи, що взаємодіє з сервером бази даних.	2
	<b>Усього годин</b>	<b>32</b>



### 5. Завдання для самостійної роботи

№ з/п	Види, зміст самостійної роботи.	Кількість годин
<b>Розділ 1. Самоаналіз</b>		
1	Рефлексія: аналіз класів. Самостійне опрацювання навчального матеріалу, що було надано під час лекційного заняття по Темі 1 із Розділу 1. Контроль: опитування під час практичного заняття, опитування під час задачі індивідуального розрахункового завдання, теоретичні питання в контрольній роботі № 1	2
2	Рефлексія: створення об'єктів, масивів, динамічних проксі. Узагальнений клас Class. Самостійне опрацювання навчального матеріалу, що було надано під час лекційного заняття по Темі 2 із Розділу 1. Контроль: опитування під час практичного заняття, опитування під час задачі індивідуального розрахункового завдання, теоретичні питання в контрольній роботі № 1	2
<b>Разом</b>		<b>4</b>
<b>Розділ 2. Персистентність</b>		
3	Серіалізація: основні поняття, інтерфейс Serializable. Використання серіалізації при створенні розподіленого програмного забезпечення. Самостійне опрацювання навчального матеріалу, що було надано під час лекційного заняття по Темі 1 із Розділу 2. Контроль: опитування під час практичного заняття, опитування під час задачі індивідуального розрахункового завдання, теоретичні питання в контрольній роботі № 1	1
4	Серіалізація. Розширений інтерфейс серіалізації Externalizable. Самостійне опрацювання навчального матеріалу, що було надано під час лекційного заняття по Темі 2 із Розділу 2. Контроль: опитування під час практичного заняття, опитування під час задачі індивідуального розрахункового завдання, теоретичні питання в контрольній роботі № 1	1
5	XML документи. DTD та XSD схеми XML документів. Java API для обробки XML документів (JAXP) . Самостійне опрацювання навчального матеріалу, що було надано під час лекційного заняття по Темі 3 із Розділу 2. Контроль: опитування під час практичного заняття, опитування під час задачі індивідуального розрахункового завдання, теоретичні питання в контрольній роботі № 1	1
6	Обробка XML документів за допомогою SAX парсеру. Самостійне опрацювання навчального матеріалу, що було надано під час лекційного заняття по Темі 4 із Розділу 2. Контроль: опитування під час практичного заняття, опитування під час задачі індивідуального розрахункового завдання, теоретичні питання в контрольній роботі № 1	2
7	Document Object Model. Обробка XML документів за допомогою DOM парсеру. Змінення DOM об'єкту. Трансформація DOM об'єкту та збереження його у файлі. Самостійне опрацювання навчального матеріалу, що було надано під час лекційного заняття по Темі 5 із Розділу 2. Контроль: опитування під час практичного заняття, опитування під час задачі індивідуального розрахункового завдання, теоретичні питання в контрольній роботі № 1	2
<b>Разом</b>		<b>7</b>
<b>Розділ 3. Компонентне та мережеве програмування</b>		
8	Компонентний підхід до створення програмних продуктів. Сучасні технології компонентно-орієнтованого програмування. Самостійне опрацювання навчального матеріалу, що було надано під час лекційного заняття по Темі 1 із Розділу 3. Контроль: опитування під час практичного заняття, опитування під час задачі індивідуального розрахункового завдання.	1

№ з/п	Види, зміст самостійної роботи.	Кількість годин
9	Компонентна технологія <i>JavaBeans</i> . Робота з властивостями та подіями. Способи забезпечення персистентності компонентів. Налаштування компонентів: створення та підключення редакторів та настроювачів компонентів. Інтегрування компонентів користувача із основними середовищами розробки. Самостійне опрацювання навчального матеріалу, що було надано під час лекційного заняття по Темі 2 із Розділу 3. Контроль: опитування під час практичного заняття, опитування під час здачі індивідуального розрахункового завдання.	2
10	Клієнт-серверна архітектура. Особливості реалізації. Особливості TCP протоколу. Використання потокових (TCP) сокетів <i>Java</i> для створення мережесих клієнт / серверних застосунків. Самостійне опрацювання навчального матеріалу, що було надано під час лекційного заняття по Темі 3 із Розділу 3. Контроль: опитування під час практичного заняття, опитування під час здачі індивідуального розрахункового завдання.	2
11	Особливості протоколу UDP. Використання Datagram сокетів <i>Java</i> для створення мережесих клієнт / серверних застосунків. Самостійне опрацювання навчального матеріалу, що було надано під час лекційного заняття по Темі 4 із Розділу 3. Контроль: опитування під час практичного заняття, опитування під час здачі індивідуального розрахункового завдання.	1
12	Особливості багатадресної передачі інформації в мережі. Використання Multicast сокетів <i>Java</i> для створення мережесих застосунків. Самостійне опрацювання навчального матеріалу, що було надано під час лекційного заняття по Темі 5 із Розділу 3. Контроль: опитування під час практичного заняття, опитування під час здачі індивідуального розрахункового завдання.	1
	<b>Разом</b>	<b>7</b>
<b>Розділ 4. Розподілене програмування</b>		
13	Основні сучасні технології розробки розподіленого програмного забезпечення. Основні особливості використання технології <i>Java RMI</i> для створення мережесих клієнт / серверних застосунків. Самостійне опрацювання навчального матеріалу, що було надано під час лекційного заняття по Темі 1 із Розділу 4. Контроль: опитування під час практичного заняття, опитування під час здачі індивідуального розрахункового завдання .	2
14	Служба RMI реєстру. Особливості розгортання RMI застосунків. Кодова база. Файли політики безпеки. Служба активації RMI об'єктів. Самостійне опрацювання навчального матеріалу, що було надано під час лекційного заняття по Темі 2 із Розділу 4. Контроль: опитування під час практичного заняття, опитування під час здачі індивідуального розрахункового завдання .	2
15	Бази даних, як невід'ємна складова розподілених систем. База даних <i>Java DB</i> . Проектування та створення простих баз даних. Основні засоби мови SQL. Самостійне опрацювання навчального матеріалу, що було надано під час лекційного заняття по Темі 3 із Розділу 4. Контроль: опитування під час практичного заняття, опитування під час здачі індивідуального розрахункового завдання .	2
16	Основи JDBC: основні інтерфейси та класи. Використання бази даних <i>Java DB</i> в за <i>Java</i> стосунках. Особливості обробки помилок. Самостійне опрацювання навчального матеріалу, що було надано під час лекційного заняття по Темі 4 із Розділу 4. Контроль: опитування під час практичного заняття, опитування під час здачі індивідуального розрахункового завдання .	2
	<b>Разом</b>	<b>8</b>
	<b>Усього годин</b>	<b>26</b>

## 6. Індивідуальні завдання

Загалом передбачено 2 обов'язкових розрахункових завдання, що виконуються студентами індивідуально. Для вибору, за наступними темами:

1. Розширення об'єктно орієнтованої системи диференціювання функцій: додавання нових можливостей до готової системи (нові власні функції; таблично задані функції, що зберігають дані в структурах даних різних типів, функції, що задані аналітично; графічне відображення функцій та похідних; створення графічного інтерфейсу користувача, т.і.).
2. Рефлексивний аналіз класів і об'єктів, створення об'єктів та робота з такими об'єктами; рефлексивне створення масивів; динамічні прокси.
3. Розширена серіалізація / десеріалізація складної системи, частина класів якої не підтримує серіалізацію.
4. Створення дерева об'єктів при отриманні інформації з XML документу за допомогою валідуючого SAX парсеру, створення відповідного DOM - об'єкту, змінення його та збереження в новому XML-файлі.
5. Розширення функціональності власних *Java Beans* компонентів: можливість бінарної серіалізації / десеріалізації та серіалізації / десеріалізації в XML файл; створення власних редакторів властивостей.
6. Створення застосунку за допомогою розроблених *Java Beans* компонентів. Створення власних налаштувачів та редакторів властивостей *JavaBeans* компонентів.
7. Розширення функціональності мережевої програми обробки інформації, що базується на TCP сокетах. Створення клієнт / серверного застосунку з власним протоколом, що реалізує найпростіший варіант віддаленого виклику методів.
8. Створення мережевого застосунку з графічним інтерфейсом користувача, що реалізує багатоадресне спілкування.
9. Створення клієнт / серверного застосунку з графічним інтерфейсом користувача, що згідно з технологією *Java RMI* виконує реєстрацію учасників конференції на сервері конференції.
10. Розширення функціональних можливостей застосунку з Завдання № 9: робота з сервером бази даних *Java DB*.
11. Розширення функціональних можливостей застосунку з Завдання № 9: ізолювання прикладного/бізнес-рівня від рівня збереження за допомогою шаблону *DAO (Data Access Object)*.
12. Створення застосунку, виконаного за технологією *RMI*, де віддалені об'єкти активуються лише за запитом від клієнта.

## 7. Методи навчання

На досягнення освітніх цілей спрямовані такі методи навчання студентів:

- практичні (використовують для пізнання дійсності, формування навичок і вмінь, поглиблення знань. Під час їх застосування використовуються такі прийоми: планування виконання завдання, постановка завдання, оперативне стимулювання, контроль і регулювання, аналіз результатів, визначення причин недоліків);
- пояснювальне-ілюстративний (використовують для викладання й засвоєння нового навчального матеріалу, фактів, підходів, оцінок, висновків тощо);

– репродуктивний (для застосування студентами вивченого на основі зразка або правила, алгоритму, що відповідає інструкціям, правилам, в аналогічних до представленого зразка ситуаціях);

Як правило лекційні та практичні заняття проводяться аудиторно. В умовах дії карантину та/або воєнного стану заняття проводяться відповідно до Наказу ректора Харківського національного університету імені В.Н. Каразіна (аудиторне або дистанційно за допомогою платформ Google Meet або Zoom).

## 8. Методи контролю

Контроль засвоєння навчального матеріалу здійснюється за допомогою:

- прийому та оцінювання завдань лабораторних робіт;
- прийому та оцінювання завдань індивідуальних обчислювальних робіт;
- проведення контрольної роботи за результатами відпрацювання основних положень навчальної програми.

## 9. Схема нарахування балів

**Розподіл балів для підсумкового семестрового контролю при проведенні залікової роботи**

Поточний контроль, самостійна робота, індивідуальні завдання																Контрольна	Самостійна робота	Індивідуальні завдання	Разом	Залікова робота	Сума
Розділ 1		Розділ 2					Розділ 3					Розділ 4									
Т 1	Т 2	Т 1	Т 2	Т 3	Т 4	Т 5	Т 1	Т 2	Т 3	Т 4	Т 5	Т 1	Т 2	Т 3	Т 4						
2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	6	8	14	60	40	100

## КРИТЕРІЇ ОЦІНКИ УСПІШНОСТІ ТА РЕЗУЛЬТАТІВ НАВЧАННЯ З КУРСУ «КРОС-ПЛАТФОРМНЕ ПРОГРАМУВАННЯ»

*Форми навчальної діяльності – максимальна кількість балів, що можна отримати протягом семестру*

1. Робота на лекційних та лабораторних заняттях – 6 балів
2. Виконання лабораторних робіт – 32 бала
3. Виконання розрахунково-графічних робіт – 14 балів
4. Виконання контрольної роботи – 8 балів
5. Залікова письмова робота – 40 балів

### Критерії оцінювання відвідування занять

*Робота на лекційних заняттях (2 бали)*

- 1,0 бал: студент відвідав 80 – 100 % лекційних занять;
- 0,8 бали: студент відвідав 60 – 79 % лекційних занять;
- 0,6 бали: студент відвідав 40 – 59 % лекційних занять;
- 0,4 бал: студент відвідав 20 – 39 % лекційних занять;
- 0,2 бали: студент відвідав 1 – 19 % лекційних занять;
- 0,0 балів: студент не відвідував лекційні заняття.

Кожна правильна відповідь на запитання самостійної роботи на лекції може бути додатково оцінена в 0.5 бали (але не більше ніж 1 бал загалом)

*Робота на лабораторних заняттях (4 бали)*

1,0 бал: студент відвідав 80 – 100 % лабораторних занять;

0,8 бали: студент відвідав 60 – 79 % лабораторних занять;

0,6 бали: студент відвідав 40 – 59 % лабораторних занять;

0,4 бал: студент відвідав 20 – 39 % лабораторних занять;

0,2 бали: студент відвідав 1 – 19 % лабораторних занять;

0,0 балів: студент не відвідував лабораторні заняття.

Кожна правильна відповідь на запитання самостійної роботи на лабораторному занятті може бути додатково оцінена в 0.5 бали (але не більше ніж 3 бали загалом)

### **Критерії оцінювання виконання лабораторних робіт**

Протягом семестру студент має виконати 9 лабораторних робіт (32 бали):

**Лабораторна робота 1 за темою *Java Reflection*** – максимальна кількість балів: 3 бали; термін виконання – 2 тижні.

**Лабораторна робота 2 за темою *Java Serialization*** – максимальна кількість балів: 3 бали; термін виконання – 2 тижні.

**Лабораторна робота 3 за темою *Створення та обробка XML документів*** – максимальна кількість балів: 3 бали; термін виконання – 2 тижні.

**Лабораторна робота 4 за темою *Мережеве програмування: TCP сокети*** – максимальна кількість балів: 4 бали; термін виконання – 2 тижні.

**Лабораторна робота 5 за темою *Мережеве програмування: UDP сокети*** – максимальна кількість балів: 4 бали; термін виконання – 2 тижні.

**Лабораторна робота 6 за темою *Мережеве програмування: Multicast сокети*** – максимальна кількість балів: 4 бали; термін виконання – 2 тижні.

**Лабораторна робота 7 за темою *Розподілене програмування: основи технології RMI*** – максимальна кількість балів: 4 бали; термін виконання – 2 тижні.

**Лабораторна робота 8 за темою *Розподілене програмування: технологія RMI. Обмін класами*** – максимальна кількість балів: 4 бали; термін виконання – 2 тижні.

**Лабораторна робота 9 за темою *Бази даних та Java: основи технології JDBC*** – максимальна кількість балів: 3 бали; термін виконання – 2 тижні.

Перед виконанням лабораторної роботи студент вивчає вимоги та технічне завдання. Після цього відбувається обговорення та перевірка того, що завдання було зрозуміле коректно. Кожен студент виконує роботу самостійно. Дозволяється сумісне обговорення роботи, але програмний код кожен студент пише власноруч. Якщо програмний код різних робіт має суттєву ступінь схожості (більше 30 % коду), то ці роботи дискваліфікуються та не оцінюються.

Захист лабораторної роботи складається з двох етапів.

Перший етап – представлення результатів: готового програмного застосування та програмних кодів у відповідності до завдання лабораторної роботи.

#### Критерії оцінювання результатів роботи:

робота була виконана у відповідності з технічним завданням за вказаний час – студент отримує 100% від максимальної кількості балів; робота була виконана у відповідності з технічним завданням с запізненням до одного тижня – студент отримує 80 % від максимальної кількості балів;

робота була виконана у відповідності з технічним завданням с запізненням більше, ніж на тиждень – студент отримує 50 % від максимальної кількості балів;

технічне завдання виконано не повністю, а на  $x$  % – студент отримує  $x$  % від тих балів, що мали б бути зараховані у відповідності до строків виконання.

Другий етап здачі лабораторної роботи – відповідь на контрольні питання. Кількість питань визначає викладач за результатами представлених результатів, але не менше ніж 3. Кожне контрольне питання оцінюється таким чином:

повна розгорнута відповідь з прикладами та додатковим матеріалом, що був опрацьований на самостійній роботі – кількість балів, що отримані на першому етапі збільшується на 5 % балів (але загальний результат не може перевищувати запланований максимальний бал за роботу);

повна, але не розгорнута відповідь – кількість балів не змінюється.

неповна відповідь, або відповідь, що містить незначні та некритичні помилки чи суперечності – кількість балів, що було отримано на попередньому етапі зменшується на 5 %;

відповідь, що містить критичні помилки, або відсутність відповіді – кількість балів, що було отримано на попередньому етапі зменшується на 10 %.

#### Критерії оцінювання виконання розрахунково-графічних робіт

Протягом семестру студент має виконати 2 розрахунково - графічні роботи (14 балів):

**Розрахунково – графічна робота 1 за темою *Базові принципи ООП*** – максимальна кількість балів: 7 балів; термін виконання – 5 тижнів.

**Розрахунково – графічна робота 2 за темою *Компонента технологія JavaBeans*** – максимальна кількість балів: 7 балів; термін виконання – 5 тижнів.

Перед виконанням розрахунково-графічної роботи студент вивчає вимоги та технічне завдання. Після цього відбувається обговорення та перевірка того, що завдання було зрозуміле коректно. Кожен студент виконує роботу та оформлює її результати індивідуально. Дозволяється сумісне обговорення роботи, але програмний код та пояснювальну записку кожен студент пише власноруч. Якщо програмний код різних робіт має суттєву ступінь схожості (більше 30 % коду), то ці роботи дискваліфікуються та не оцінюються.

Захист розрахунково-графічної роботи складається з трьох етапів.

Перший етап – представлення результатів: готового програмного рішення та програмних кодів у відповідності до завдання лабораторної роботи. (5 балів)

Критерії оцінювання результатів роботи:

робота була виконана у відповідності з технічним завданням за вказаний час – студент отримує 100% від максимальної кількості балів; робота була виконана у відповідності з технічним завданням с запізненням до одного тижня – студент отримує 80 % від максимальної кількості балів;

робота була виконана у відповідності з технічним завданням с запізненням більше, ніж на тиждень – студент отримує 50 % від максимальної кількості балів;

технічне завдання виконано не повністю, а на  $x$  % – студент отримує  $x$  % від тих балів, що мали б бути зараховані у відповідності до строків виконання.

Другий етап задачі розрахунково-графічної роботи – відповідь на контрольні питання. Кількість питань визначає викладач за результатами представлених результатів, але не менше ніж 3. Кожне контрольне питання оцінюється таким чином:

повна розгорнута відповідь з прикладами та додатковим матеріалом, що був опрацьований на самостійній роботі – кількість балів, що отримані на першому етапі збільшується на 5 % балів (але загальний результат не може перевищувати запланований максимальний бал за роботу) ;

повна, але не розгорнута відповідь – кількість балів не змінюється.

неповна відповідь, або відповідь, що містить незначні та некритичні помилки чи суперечності – кількість балів, що було отримано на попередньому етапі зменшується на 5 %;

відповідь, що містить критичні помилки, або відсутність відповіді – кількість балів, що було отримано на попередньому етапі зменшується на 10 %.

Третій етап задачі розрахунково-графічної роботи – перевірка оформлення пояснювальної записки до розрахунково – графічної роботи. (2 бали).

Критерії оцінювання результатів роботи:

Робота виконана повністю у відповідності із *Рекомендаціями щодо оформлення розрахунково-графічних робіт студентами II курсу факультету комп'ютерних наук з дисципліни «крос-платформне програмування»* - 2 бали

Незначні помилки в оформленні роботи (всі розділи присутні; є діаграми, що показують структуру програмного рішення, в наявності результати тестування; результати роботи адекватно представлені але є помилки в оформленні, або недостатньо представлені або структура програмного рішення, або результати тестування, або результати роботи) – 1.5-1.9 балів

Значні помилки в оформленні роботи (не всі розділи присутні; або відсутні діаграми, що показують структуру програмного рішення, або результати тестування; або результати роботи представлені не достатньо) – 1.0-1.4 балів

Декілька значних помилок в оформленні роботи (не всі розділи присутні та або відсутні діаграми, що показують структуру програмного рішення, або результати тестування; або результати роботи представлені не достатньо) – 0.5-0.9 балів

Велика кількість суттєвих помилок (не всі розділи присутні та відсутні діаграми, що показують структуру програмного рішення та результати тестування та результати роботи представлені не достатньо) – *0-0.4 бали*

Пояснювальна записка не надана – робота не приймається, бали що отримані на попередніх етапах не нараховуються.

### **Критерії оцінювання виконання контрольних робіт**

Протягом семестру студент має виконати одну письмову контрольну роботу. Контрольні роботи виконуються в аудиторії. Вона розрахована на одну *50* хвилин та оцінюється у *8 балів*.

Кожна контрольна робота складається з завдань трьох типів:

1. завдання з вибором відповіді з переліку запропонованих – студент має вибрати правильний варіант (або декілька варіантів в залежності від питання) відповіді серед декількох запропонованих;
2. завдання з короткою вільною відповіддю;
3. завдання з розгорнутою вільною відповіддю.

Завдання першого типу оцінюються:

100% від запланованої кількості балів (*0.5 бала*) у випадку правильної відповіді;

0 балів – у випадку помилки, або відсутності відповіді.

Завдання другого типу оцінюються таким чином:

100% від запланованої кількості балів (*1.0 бал*) у випадку правильної відповіді;

80% від запланованої кількості балів – у випадку незначної неточності у відповіді;

50% від запланованої кількості балів – у випадку декількох незначних неточностей у відповіді;

0 балів – у випадку значної неточності, помилки, або відсутності відповіді.

Завдання третього типу оцінюються таким чином:

правильна, повна відповідь – 100 % від запланованої кількості балів (*2.0 бали*) ;

повна відповідь, що містить незначні помилки або неточності чи суперечність – 80 % від запланованої кількості балів;

неповна відповідь, що не містить критичних помилок чи суперечностей – 60 % від запланованої кількості балів;

неповна відповідь в великою кількістю помилок – 40 % від запланованої кількості балів;

відповідь, що містить критичну помилку, або відсутність відповіді оцінюється в 0 балів.

### **Критерії оцінювання виконання залікової письмової роботи**

В кінці семестру студент має виконати письмову залікову роботу. Ця робота виконуються в аудиторії. Вона розрахована на 1 пару (80 хвилин), складається з декількох питань, на які треба дати вільну розгорнуту відповідь, та оцінюються у 40 балів. Кожне питання має фіксовану максимальну кількість балів та оцінюється окремо таким чином:



- повна, розгорнута відповідь, що містить приклади та додаткові відомості, які були опрацьовані протягом семестру на самостійній роботі – 100 % від запланованої кількості балів за питання;
- повна, але не розгорнута відповідь, відсутність достатньої кількості прикладів – 90 % від запланованої кількості балів за питання;
- повна, але не розгорнута відповідь, що містить незначні помилки чи неточності – 80 % від запланованої кількості балів за питання;
- неповна відповідь, що не містить критичних помилок та неточностей – 70 % від запланованої кількості балів за питання;
- неповна відповідь, що містить істотні, але не критичні помилки або неточностей – 50 % від запланованої кількості балів за питання;
- відповідь, що в цілому вірна, але містить критичну помилку чи неточність – 25 % від запланованої кількості балів за питання;
- відповідь не вірна, не дана на поставлене запитання або відсутність відповіді оцінюється в 0 балів.

### Шкала оцінювання

Сума балів за всі види навчальної діяльності протягом семестру	Оцінка	
	для чотирирівневої шкали оцінювання	для дворівневої шкали оцінювання
90 – 100	відмінно	зараховано
70-89	добре	
50-69	задовільно	
1-49	незадовільно	не зараховано

## 10. Рекомендована література

### Основна література

1. Horstmann C. S. Core Java, Vol. I: Fundamentals. 12th Edition: – Oracle Press, 2021 – 1197 p.
2. Horstmann C. S. Core Java, Vol. II: Advanced Features. 12th Edition: – Oracle Press, 2022 – 1192 p.
3. Uttam Kumar Roy. Advanced Java Programming: - OUP India; UK ed. Edition, 2015 – 880 p.
4. Loy M., Niemeyer P., Leuck D. Learning Java. An Introduction to Real-World Programming with Java. 5 ed: - O'Reilly Media, Inc., 2020. – 518 p.
5. Eckel B. Thinking in Java. 4th Edition: - Pearson, 2006.- 1150 c.

### Допоміжна література

1. Wang A. J., Qian K. Component-oriented programming: - John Wiley & Sons. Inc., 2005.–334p.
2. Forman I.R., Forman N. Java Reflection in Action: - Manning Publications Co., 2005 – 297 p.
3. Harold E. R. Java I/O: - O'Reilly, 1999 – 596 p.
4. Harold E.R., Means S.W. XML in a Nutshell. - O'Reilly Media, 2002 – 640 c.
5. Englander R. Developing JAVA Beans: - O'Reilly, 1997 – 231 p.
6. Harold E. R. Java Network Programming: - O'Reilly, 2005 – 735 p.

7. Grosso W. Java RMI: - O'Reilly, 2001 – 572 p.
8. Doss G. M. CORBA Networking with Java: - Wordware Publishing, 1999 – 327 p.
9. Kishori Sharan. Java APIs, Extensions and Libraries: With JavaFX, JDBC, jmod, jlink, Networking, and the Process API 2nd Edition: - Apress, 2018 – 838 p.
10. Ying Bai. Practical Database Programming with Java: - Wiley-IEEE Press, 2011 – 952 p.
11. C. J. Date. Database Design and Relational Theory: Normal Forms and All That Jazz, 2nd Edition: - Apress, 2019 – 470 p.
12. C. J. Date. SQL and Relational Theory: How to Write Accurate SQL Code, 2nd Edition: - O'Reilly Media, 2012 – 448 p.
13. Weisfeld M. The Object-Oriented Thought Process, 5th Edition: - Addison-Wesley, 2019 – 336 p.
14. Joshua Bloch. Effective Java 3rd Edition: - Addison-Wesley Professional, 2017 - 416 p.
15. Cay S. Horstmann. Object-Oriented Design and Patterns 2nd Edition: - Wiley, 2005 – 450 p.
16. Vaskaran Sarcar. Java Design Patterns: A Hands-On Experience with Real-World Examples 3rd Edition: - Apress, 2022 - 696 p.

### **11. Посилання на інформаційні ресурси в Інтернеті, відео-лекції, інше методичне забезпечення**

1. Java Platform Standard Edition 8 Documentation: [Електронний ресурс]// Oracle and/or its affiliates, 1993-2017. URL: <https://docs.oracle.com/javase/8/docs>.
2. Java API for XML Processing (JAXP): [Електронний ресурс]// Oracle and/or its affiliates, 1993-2017. URL: <https://docs.oracle.com/javase/tutorial/jaxp/>
3. Processing XML with Java: [Електронний ресурс]// Elliott Rusty Harold, 2001-2003. URL: <http://www.cafeconleche.org/books/xmljava/>
4. XML Technology: [Електронний ресурс]// W3C, 2015. URL: <http://www.w3.org/standards/xml/>
5. DTD Schema: [Електронний ресурс]// W3C, 2015. URL: <http://www.w3schools.com/DTD/>
6. Namespaces in XML: [Електронний ресурс]// W3C, 2015.. URL: <http://www.w3.org/TR/xml-names/>
7. XML Schema: [Електронний ресурс]// W3C, 2015. URL: <http://www.w3.org/XML/Schema.html>
8. Document Object Model (DOM): [Електронний ресурс]// W3C, 2015. URL: <http://www.w3.org/DOM/>
9. Trail: JavaBeans(TM): [Електронний ресурс]// Oracle and/or its affiliates, 1993-2017. URL: <https://docs.oracle.com/javase/tutorial/javabeans/>
10. JavaBeans: [Електронний ресурс]// Oracle and/or its affiliates, 1993-2017. URL: <http://java.sun.com/products/javabeans>
11. Trail: Custom Networking: [Електронний ресурс]// Oracle and/or its affiliates, 1993-2017. URL: <https://docs.oracle.com/javase/tutorial/networking/>
12. Trail: RMI: [Електронний ресурс]// Oracle and/or its affiliates, 1993-2017. URL: <https://docs.oracle.com/javase/tutorial/rmi/>
13. Java Distributed Computing: [Електронний ресурс]// O'Reilly & Associates, 2001. URL: <http://docstore.mik.ua/orelly/java-ent/dist/>

14. Java Enterprise in Nutshell: [Электронный ресурс]// O'Reilly & Associates, 2001. URL: <http://docstore.mik.ua/oreilly/java-ent/jenut/index.htm>
15. Enterprise Java Beans: [Электронный ресурс]// O'Reilly & Associates, 2001. URL: <http://docstore.mik.ua/oreilly/java-ent/ebeans/index.htm>
16. Java Programming Fundamentals: [Электронный ресурс]// EdX & GalileoX – 2023. URL: <https://www.edx.org/course/java-programming-fundamentals>.
17. Distributed Systems using Java: [Электронный ресурс]// Gardenia Tech – 2020. URL: <https://www.youtube.com/playlist?list=PLtDIUAtyP4lhV7CsYfLuIx26UeG4J-ujZ>.
18. Jenkov Tech & Media Labs: [Электронный ресурс]// jenkov.com, 2023. URL: <https://jenkov.com/>