



КРОСПЛАТФОРМНЕ ПРОГРАМУВАННЯ

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	12 Інформаційні технології
Спеціальність	121 Інженерія програмного забезпечення
Освітня програма	Інженерія програмного забезпечення мультимедійних та інформаційно-пошукових систем
Статус дисципліни	Вибіркова
Форма навчання	Очна (денна)
Рік підготовки, семестр	3 рік підготовки, 6 семестр
Обсяг дисципліни	Лекції: 36 год., лабораторні роботи: 18 год., самостійна робота: 66 год.
Семестровий контроль/ контрольні заходи	Залік, модульна контрольна робота, календарний контроль
Розклад занять	Згідно розкладу на весняний семестр поточного навчального року (rozklad.kpi.ua)
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: к.ф.-м.н., доцент, Ткаченко О. А., aatokq@gmail.com Лабораторні роботи: к.ф.-м.н., доцент, Ткаченко О. А., aatokq@gmail.com
Розміщення курсу	Google classroom. Доступ надається зареєстрованим студентам.

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Вивчення дисципліни «Кросплатформне програмування» дозволяє сформувати у здобувачів освіти компетенції, необхідні для розв'язання практичних задач професійної діяльності, пов'язаної із розробленням кросплатформного програмного забезпечення.

Метою вивчення дисципліни «Кросплатформне програмування» є формування у здобувачів освіти здатностей самостійно проєктувати та розробляти кросплатформне програмне забезпечення.

Предметом дисципліни «Кросплатформне програмування» є методи, засоби моделювання і розроблення кросплатформного програмного забезпечення.

Вивчення дисципліни «Кросплатформне програмування» сприяє формуванню у здобувачів освіти **фахових компетентностей (ФК)**, необхідних для розв'язання практичних задач професійної діяльності, пов'язаної з розробленням, вдосконаленням та експлуатуванням програмного забезпечення:

ФК01 Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.

ФК02 Здатність брати участь у проєктуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

ФК03 Здатність розробляти архітектури, модулі та компоненти програмних систем

ФК05 Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу.

ФК06 Здатність аналізувати, вибирати і застосовувати методи і засоби для забезпечення інформаційної безпеки (в тому числі кібербезпеки).

ФК11 Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розроблення програмного забезпечення.

ФК14 Здатність до алгоритмічного та логічного мислення.

ФК21 Здатність визначати, аналізувати та документувати вимоги до програмного забезпечення мультимедійних та інформаційно-пошукових систем.

Вивчення дисципліни «Кросплатформне програмування» сприяє формуванню у студентів наступних **програмних результатів навчання (ПРН)** за освітньою програмою:

ПРН03 Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.

ПРН05 Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розроблення програмного забезпечення.

ПРН06 Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.

ПРН09 Вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.

ПРН10 Проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування.

ПРН11 Вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання.

ПРН12 Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.

ПРН15 Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.

ПРН17 Вміти застосовувати методи компонентної розробки програмного забезпечення.

ПРН21 Знати засоби, аналізувати, вибирати, кваліфіковано застосовувати засоби забезпечення інформаційної безпеки (в тому числі кібербезпеки) і цілісності даних відповідно до розв'язуваних прикладних завдань та створюваних програмних систем.

ПРН23 Вміти документувати та презентувати результати розроблення програмного забезпечення.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Успішному вивченню дисципліни «Кросплатформне програмування» передують вивчення дисциплін «Основи програмування», «Алгоритми та структури даних», «Основи комп'ютерних систем і мереж», «Компоненти програмної інженерії», «Бази даних» та «Програмування» навчального плану підготовки бакалаврів за спеціальністю 121 Інженерія програмного забезпечення.

Отримані при засвоєнні дисципліни «Кросплатформне програмування» теоретичні знання та практичні уміння забезпечують успішне виконання курсових проєктів та магістерських дисертацій за спеціальністю 121 Інженерія програмного забезпечення.

3. Зміст навчальної дисципліни

Дисципліна «Кросплатформне програмування» передбачає вивчення таких тем:

Тема 1. Вимоги до кросплатформного програмного забезпечення. Основи мови java.

Тема 2. Реалізація принципів об'єктно-орієнтованого програмування у мові Java.

Тема 3. Робота з класами та об'єктами. Колекції.

Тема 4. Розробка графічного інтерфейсу користувача. Організація багатопоточності.

Модульна контрольна робота

4. Навчальні матеріали та ресурси

Базова література

1. Кетті Сьєрра, Берт Бейтс. *Head First Java. Легкий для сприйняття довідник. 2 видання оновлене.* – Харків: Фабула, 2022. – 720 с.
2. Васильєв О. *Програмування мовою Java.* – Київ: Навчальна література – Богдан, 2020. – 696 с.
3. Ерік Фрімен, Елізабет Робсон. *Патерни проєктування.* – Харків: Фабула, 2020. – 672 с.
4. Тарнавський Ю. А. *Java-програмування: комп'ютерний практикум: навч. посіб.* Київ: КПІ ім. Ігоря Сікорського, 2021. – 95 с.
5. Ткаченко О.А., Ткаченко О.І., Овчарук І.В. *Сучасні парадигми програмування. Навч. посіб. Ч.2.* – Київ: КНУКіМ, 2017. – 305с.
6. Мельник Р.А. *Програмування веб-застосувань (фронт-енд та бек-енд).* – Львів: Львівська політехніка, 2018. – 244 с.
7. Горбань А.Г. *Програмування в Java: Курс лекцій.* – Полтава: Полтавський державний технічний університет, 2008. – 310 с.
8. В'язовик М.О. *Програмування на Java [Курс лекцій].* – INTUIT, 2003. – 391 с.
9. Копитко М.Ф., Іванків К.С. *Основи програмування мовою Java: Тексти лекцій.* – Львів: Видавничий центр ЛНУ ім. Івана Франка, 2002.– 83 с.
10. Брнакевич І.Є., Вагін П.П. *Програмування мовою Java: використання фундаментальних класів: Тексти лекцій.* – Львів: Видавничий центр ЛНУ імені Івана Франка, 2002. – 75 с.

Додаткова література:

11. *Software Development Studio: Guide to Laboratory Work/O.Tkachenko* – К.: НАУ, 2021. – 32 р.
12. C. Ullenboom. *Java: The Comprehensive Guide to Java Programming for Professionals.* – Rheinwerk Computing, 2022. – 1128р.
13. Y. Daniel Liang. *Introduction to Java Programming and Data Structures.* – Pearson, 2020. – 1240р.
14. Downey A.B., Mayfield C. *Think Java: How to Think Like a Computer Scientist Version 6.1.3* /A.B.Downey, C. Mayfield. – Massachusetts: Green Tea Press, 2016. – 291р.
15. Evans B. *Java: The Legend* /B. Evans. – Publisher: O'Reilly Media, 2015. – 769р.
16. Aniche M. *Effective Software Testing: A developer's guide.* Shelter Island, USA: Manning, 2022. – 328р.
17. Dennis A., Wixom B., Tegarden D. *Systems Analysis and Design : An Object-Oriented Approach with UML.* Hoboken, USA : Wiley, 2020. 544 р.
18. www.java.com
19. <https://docs.oracle.com/javase/tutorial/>
20. <https://www.cs.cmu.edu/afs/cs.cmu.edu/user/gchen/www/download/java/LearnJava.pdf>
21. <https://book.huihoo.com/goalkicker.com/JavaBook/JavaNotesForProfessionals.pdf>
22. <https://introcs.cs.princeton.edu/java/10elements/>
23. <https://www.oracle.com/java/technologies/>

Використати для опанування практичних умінь дисципліни. Матеріали знаходяться у вільному доступі в Інтернеті.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

№ з/п	Тип навчального заняття	Опис навчального заняття
<i>Тема 1. Вимоги до кросплатформного програмного забезпечення. Основи мови java.</i>		
1	<i>Лекція 1. Зміст курсу, вступ до кросплатформного програмування</i>	<i>Огляд змісту курсу. Поняття кросплатформності як властивості програмного забезпечення. Різновиди кросплатформності програмного забезпечення. Життєвий цикл розроблення кросплатформного програмного забезпечення. Способи та засоби досягнення кросплатформності програмного забезпечення. Завдання на СРС: п. 6 № 1.</i>
2	<i>Лекція 2. Основи мови програмування JAVA</i>	<i>Загальна характеристика мови Java. Історія створення мови JAVA. Відмінності мов JAVA та C++. Лексика та синтаксис мови. Прості типи даних. Класи – оболонки типів, їхні методи. Визначення локальних змінних. Консольне введення/виведення. Структура програми на мові JAVA. Завдання на СРС: п. 6 № 2.</i>
3	<i>Лекція 3. Операції та оператори мови. Робота з масивами та рядками.</i>	<i>Арифметичні та логічні операції. Приоритети виконання операцій. Оператори управління виконанням програми. Робота з масивами. Класи та методи роботи з рядками. Класи String, StringBuilder, StringBuffer, StringTokenizer. Завдання на СРС: п. 6 № 3.</i>
4	<i>Лабораторна робота 1. Робота з багатовимірними масивами</i>	<i>Завдання: ознайомитися с базовими елементами мови Java та навчитися оброблювати набори даних у вигляді масивів. Завдання на СРС: п. 6 № 4.</i>
5	<i>Лекція 4. Обробка виняткових ситуацій.</i>	<i>Обробка виняткових ситуацій. Поняття виключення. Типи виключень. Розділи try, catch та finally. Оператор throw. Ключове слово throws. Особливості обробки виключень в перевизначених методах. Множинний обробник виключень. Умова об'єднання виключень. Створення власного виключення. Завдання на СРС: п. 6 № 5.</i>
6	<i>Лабораторна робота 2. Робота з рядками символів</i>	<i>Завдання: вивчити можливості мови Java для обробки текстових типів даних. Завдання на СРС: п. 6 № 6.</i>

<i>Тема 2. Реалізація принципів об'єктно-орієнтованого програмування у мові Java</i>		
7	<i>Лекція 5. Класи та об'єкти. Реалізація принципів об'єктно-орієнтованого програмування.</i>	<i>Класи та об'єкти. Спадкування, інкапсуляція та поліморфізм. Посилання. Збирання сміття. Методи і конструктори. Вбудовані методи. Передача параметрів методам. Суміщення (перевантаження) та заміщення методів і конструкторів. Завдання на СРС: п. 6 № 7.</i>
8	<i>Лекція 6. Пакети. Абстрактні класи та інтерфейси</i>	<i>Пакети. Абстрактні класи та методи, їх особливості. Інтерфейси. Функціональні інтерфейси. Реалізація Інтерфейсу. Змінні інтерфейсу в якості констант. Реалізація множинного наслідування. Реалізація поліморфізму. Завдання на СРС: п. 6 № 8.</i>
9	<i>Лабораторна робота 3. Реалізація принципів об'єктно-орієнтованого програмування</i>	<i>Завдання: вивчити реалізацію принципів ООП (успадкування і поліморфізм), а також механізми взаємодії між класами. Завдання на СРС: п. 6 № 9.</i>
10	<i>Лекція 7. Потоки виконання – базові засоби.</i>	<i>Поняття потоку виконання. Створення потоку - клас Thread. Створення потоку - інтерфейс Runnable. Пріоритети потоків. Синхронізація потоків. Взаємодія потоків. Механізм асинхронного виконання. Завдання на СРС: п. 6 № 10.</i>
11	<i>Лекція 8. Робота с базами даних</i>	<i>ODBC і JDBC. Створення бази даних та таблиць. Встановлення з'єднання з БД. Створення та виконання запитів. Обробка результатів запиту. Завдання на СРС: п.6 № 11.</i>
12	<i>Лабораторна робота 4. Дослідження механізму обробки виключень при роботі з класами</i>	<i>Завдання: Вивчити поняття виключення та механізм обробки виключень в Java. Дослідити створення власних типів виключень. Завдання на СРС: п.6 № 12.</i>
13	<i>Лекція 9. Потокове введення / виведення даних</i>	<i>Потоки для різних типів даних. Потоки байтового та символьного введення/виведення. Файлові потоки введення/виведення. Буферизоване введення/виведення символів та рядків. Потоки для роботи з примітивними типами. Об'єктні потоки. Завдання на СРС: п. 6 № 13.</i>
14	<i>Лекція 10. Вкладені класи</i>	<i>Типи вкладених класів. Статичний клас. Внутрішній та локальний клас. Анонімний локальний клас. Використання вкладених класів. Обробка вкладених класів компілятором та віртуальною машиною. Завдання на СРС: п.6 № 14.</i>

Тема 3. Робота з класами та об'єктами. Колекції.

15	Лекція 11. Контейнерні класи. Узагальнення	Класи-контейнери. Базові інтерфейси контейнерних класів: <i>Collection</i> і <i>Map</i> . Класи-перелічення. Поняття узагальнення. Узагальнений метод. Обмеження на параметр типу. Обмеження на узагальнення. Узагальнення і спадкування. Підстановлювальний тип. Завдання на СРС: п. 6 № 15.
16	Лекція 12. Сериалізація	Сериалізація і десериалізація. Сериалізація складних об'єктів. Інтерфейс <i>java.io.Serializable</i> . Статичне поле <i>serialVersionUID</i> . Класи <i>ObjectOutputStream</i> та <i>ObjectInputStream</i> . Сериалізація і спадкування. Користувальницька сериалізація. Завдання на СРС: п. 6 № 16.
17	Лабораторна робота 5. Дослідження вкладених класів	Завдання: Вивчити поняття вкладений клас, типи вкладених класів (статичні, внутрішні, локальні та анонімні). Дослідити правила використання різних типів вкладених класів. Завдання на СРС: п.6 № 17.
18	Лекція 13. Колекції	Інтерфейс <i>Collection</i> та клас <i>Collections</i> . Інтерфейс <i>Iterator</i> . Спискові інтерфейси: <i>List</i> , <i>Queue</i> , <i>Deque</i> . Множина <i>Set</i> . Карти відображень <i>Map</i> . Класи <i>HashSet</i> , <i>LinkedHashSet</i> і <i>HashMap</i> . Клас <i>TreeSet</i> Реалізації колекцій, безпечних з точки зору потоків управління. Завдання на СРС: п. 6 № 18.
19	Лекція 14. Завантажувачі класів. Рефлексія	Завантажувачі класів. Модель делегування завантаження класів. Призначення рефлексії. Способи отримання об'єкту типу <i>Class</i> . Отримання інформації про модифікатори типу даних. Отримання інформації про елемент класу. Динамічний виклик методів. Доступ до приватних членів типу даних. Завдання на СРС: п. 6 № 19.
20	Лабораторна робота 6. Дослідження фреймворку колекцій	Завдання: Ознайомитись з фреймворком колекцій мови <i>java</i> , дослідити від чого залежить ефективність обробки колекцій та які колекції потрібно використовувати в залежності від операцій пошуку, додавання та видалення. Завдання на СРС: п.6 № 20.

<i>Тема 4. Розробка графічного інтерфейсу користувача. Організація багатопоточності</i>		
21	<i>Лекція 15. Розробка графічного інтерфейсу користувача</i>	<i>Стандартні бібліотеки Java для розробки графічного інтерфейсу користувача. Переваги бібліотеки JavaFX у порівнянні з AWT та Swing. Розміщення компонентів. Обробка подій. Завдання на СРС: п. 6 № 21.</i>
22	<i>Лекція 16. Багатопотокове програмування - пакет java.util.concurrent</i>	<i>Переваги пакету java.util.concurrent. Інтерфейси Lock та Condition. Інтерфейс Callable. Пул потоків та виконавці. Блокуючі черги. Синхронізуючі класи-колекції. Синхронізатори загального призначення. Атомарні класи та змінні. Завдання на СРС: п. 6 № 22.</i>
23	<i>Модульна контрольна робота.</i>	<i>Завдання: продемонструвати робочий проект кросплатформного програмного забезпечення. Завдання на СРС: п.6 № 23.</i>
24	<i>Лекція 17. Stream API. Lambda-вирази</i>	<i>Концепція потоку stream. Способи створення стрімів. Методи роботи зі стрімами. Концепція Lambda-виразу. Використання stream з Lambda-виразами. Завдання на СРС: п. 6 № 24.</i>
25	<i>Лекція 18. Тестування кросплатформного програмного забезпечення</i>	<i>Поняття тестування кросплатформного програмного забезпечення. Життєвий цикл кросплатформного програмного забезпечення. Методи тестування кросплатформного програмного забезпечення. Особливості тестування кросплатформного програмного забезпечення. Завдання на СРС: п. 6 № 25.</i>
26	<i>Лабораторна робота 7. Дослідження організації багатопоточності</i>	<i>Завдання: вивчити механізми створення та управління потоками виконання в Java засобами пакету java.util.concurrent; дослідити способи синхронізації потоків; ознайомитися з додатковими можливостями щодо синхронізації та блокування доступу до ресурсів. Завдання на СРС: п. 6 № 26.</i>
27	<i>Залік</i>	<i>Завдання на СРС: п. 6 № 27.</i>

6. Самостійна робота студента/аспіранта

Дисципліна «Кросплатформне програмування» ґрунтується на самостійних підготовках до аудиторних занять на теоретичні та практичні теми.

<i>№ з/п</i>	<i>Назва теми, що виноситься на самостійне опрацювання</i>	<i>Кількість годин</i>	<i>Література</i>
1	<i>Підготовка до лекції 1</i>	1,5	5; 12; 13, 17

2	Підготовка до лекції 2	1,5	1-4; 7-10
3	Підготовка до лекції 3	1,5	1-4; 7-10
4	Підготовка до лабораторної роботи 1	4	1-4, 11
5	Підготовка до лекції 4	1,5	1-5; 12-15; 17
6	Підготовка до лабораторної роботи 2	4	1-4; 11
7	Підготовка до лекції 5	1,5	1-5; 12-17
8	Підготовка до лекції 6	1,5	7-10; 12-15
9	Підготовка до лабораторної роботи 3	4	4, 5, 7, 9, 11
10	Підготовка до лекції 7	1,5	7-9; 12-15
11	Підготовка до лекції 8	1,5	1-8; 12-15
12	Підготовка до лабораторної роботи 4	4	1-4; 7-10; 11
13	Підготовка до лекції 9	1,5	1-5; 12-17
14	Підготовка до лекції 10	1,5	1-5; 12-17
15	Підготовка до лекції 11	1,5	1-5; 8; 9
16	Підготовка до лекції 12	1,5	1-5; 14-16
17	Підготовка до лабораторної роботи 5	4	1-4; 7-10; 11
18	Підготовка до лекції 13	1,5	1-3; 8; 9, 18-23
19	Підготовка до лекції 14	1,5	1-3; 8; 9, 18-23
20	Підготовка до лабораторної роботи 6	4	1-4; 11; 12-15
21	Підготовка до лекції 15	1,5	1-3; 12-15
22	Підготовка до лекції 16	1,5	1-3; 12-15
23	Підготовка до модульної контрольної роботи	6	1-23
24	Підготовка до лекції 17	1,5	3, 5, 7, 12-17
25	Підготовка до лекції 18	1,5	5, 15, 16, 18-23
26	Підготовка до лабораторної роботи 7	3	1-4; 11; 12-15
27	Підготовка до заліку	6	1-23

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Відвідування лекційних занять є обов'язковим.

Відвідування занять лабораторних робіт може бути епізодичним та за потреби консультації/захисту лабораторних робіт.

Правила поведінки на заняттях: активність, повага до присутніх, відключення телефонів.

Дотримання політики академічної доброчесності.

Правила захисту лабораторних робіт: роботи повинні бути виконані відповідно до поставлених завдань та згідно з обраним студентом варіантом.

8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Протягом семестру студенти виконують 7 лабораторних робіт

Максимальна кількість балів за кожну лабораторну роботу 10 балів.

Бали нараховуються за якість виконання і захист лабораторних робіт: 0-10 балів.

Критерії оцінювання якості виконання і захисту:

- якість виконання роботи: 0-7 бали;

- своєчасне представлення роботи до захисту: 0-3 бали.

Критерії оцінювання якості виконання:

7 балів – робота виконана якісно, в повному обсязі;

5-6 бали – робота виконана якісно, в повному обсязі, але має недоліки;

3-4 бали – робота виконана в повному обсязі, але містить незначні помилки;

1-2 бали – робота виконана в повному обсязі, але містить суттєві помилки;

0 балів - робота виконана не якісно, не в повному обсязі, відповідей або немає, або невірні.

Критерії оцінювання своєчасності представлення роботи до захисту:

3 бали – робота представлена до захисту не пізніше вказаного терміну;

1 бал – робота представлена до захисту на 1 тиждень пізніше вказаного терміну;

0 балів – робота представлена до захисту більш ніж 2 тижні пізніше вказаного терміну.

Максимальна кількість балів за виконання та захист лабораторних робіт:

10 балів x 7 лабораторних робіт = 70 балів.

Завдання на **модульну контрольну роботу** складається з 30 тестових запитань. Відповідь на кожне запитання оцінюється 1 балом.

Критерії оцінювання запитання контрольної роботи:

1 бал – відповідь вірна;

0 балів – немає відповіді або відповідь невірна.

Максимальна кількість балів за модульну контрольну роботу:

1 бал x 30 запитань = 30 балів.

Рейтингова шкала з дисципліни дорівнює:

$R = RC = 70 \text{ балів} + 30 \text{ балів} = 100 \text{ балів}$.

Календарний контроль: проводиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу.

На першій атестації (8-й тиждень) студент отримує «Зараховано», якщо його поточний рейтинг не менше 15 балів (50 % від максимальної кількості балів, яку може отримати студент до першої атестації).

На другій атестації (14-й тиждень) студент отримує «Зараховано», якщо його поточний рейтинг не менше 35 балів (50 % від максимальної кількості балів, яку може отримати студент до другої атестації).

Семестровий контроль: залік

Умови допуску до семестрового контролю:

При семестровому рейтингу (R_c) не менше 60 балів студент отримує залік «автоматом» відповідно до таблиці (Таблиця відповідності рейтингових балів оцінкам за університетською

шкалою). В іншому випадку ним виконується залікова контрольна робота.

Якщо студент не погоджується з оцінкою «автоматом», то може спробувати підвищити свою оцінку шляхом написання залікової контрольної роботи. При цьому його бали, отримані за семестр, зберігаються, а з двох отриманих студентом оцінок виставляється краща («м'яка» система оцінювання).

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

9. Додаткова інформація з дисципліни (освітнього компонента)

Перелік питань, які виносяться на семестровий контроль, наведено у Додатку 1.

Робочу програму навчальної дисципліни (силабус):

Складено к.ф.-м.н., доцент Ткаченко О.А.

Ухвалено кафедрою ПЗКС (протокол №8 від 25.01.23)

Погоджено Методичною комісією факультету прикладної математики (протокол №6 від 27.01.23)

Додаток 1. Перелік питань, які виносяться на семестровий контроль

- 1. Поняття кросплатформності як властивості програмного забезпечення.*
- 2. Різновиди кросплатформності програмного забезпечення.*
- 3. Засоби досягнення кросплатформності програмного забезпечення.*
- 4. Методи визначення функційних вимог до кросплатформного програмного забезпечення.*
- 5. Реалізація кросплатформності мови java. Відмінності мови Java від C++.*
- 6. Середовища розробки застосунків (IDE).*
- 7. Примітивні типи даних. Класи-оболонки типів даних.*
- 8. Літерали, змінні, вбудовані функції.*
- 9. Оператори.*
- 10. Масиви.*
- 11. Засоби роботи з символами та рядками.*
- 12. Обробка виняткових ситуацій.*
- 13. Обробка текстових та бінарних файлів.*
- 14. Пакети.*
- 15. Інкапсуляція. Класи та об'єкти класів. Специфікатори доступу.*
- 16. Конструктори класів.*
- 17. Методи. Способи передачі параметрів.*
- 18. Спадкування: засоби реалізації.*
- 19. Поліморфізм: засоби реалізації.*
- 20. Суміщення (перевантаження) та заміщення методів.*
- 21. Абстрактні методи та класи. Інтерфейси.*
- 22. Внутрішні класи.*
- 23. Узагальнені типи.*
- 24. Колекції. Перелічення.*
- 25. Створення графічного інтерфейсу користувача.*
- 26. Stream API. Lambda-вирази.*
- 27. Серіалізація.*
- 28. Рефлексія.*
- 29. Організація багатопотоковості.*
- 30. Методи тестування кросплатформного програмного забезпечення.*