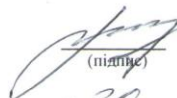


Міністерство освіти і науки України
Національний аерокосмічний університет ім. М. Є. Жуковського
«Харківський авіаційний інститут»

Кафедра інженерії програмного забезпечення (№ 603)

ЗАТВЕРДЖУЮ

Керівник проектної групи/

І.Б. Туркін
(ініціали та прізвище)
«30» 08 2019 р.

**РОБОЧА ПРОГРАМА ВИБІРКОВОЇ
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Крос-платформне програмування
(назва навчальної дисципліни)

Галузь знань: 12 «Інформаційні технології»

(шифр і найменування галузі знань)

Спеціальність: 121 «Інженерія програмного забезпечення»

(код та найменування спеціальності)

Освітня програма: «Інженерія програмного забезпечення»

(найменування освітньої програми)

Форма навчання: денна


Рівень вищої освіти: перший (бакалаврський)

Харків 2019 рік

Робоча програма «Крос-платформне програмування» для студентів за спеціальністю: 121 «Інженерія програмного забезпечення» освітньою програмою «Інженерія програмного забезпечення»

«20» 04 2019 р, – 16 с.

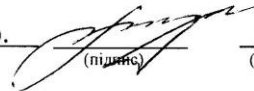
Розробник: Резуненко О. О., ст. викладач кафедри №603,
(прізвище та ініціали, посада, науковий ступінь та вчене звання)


(підпис)

Робочу програму розглянуто на засіданні кафедри інженерії програмного забезпечення

Протокол № 1 від «30» 08 2019 р.

Завідувач кафедри д-р техн. наук., проф.
(науковий ступінь і вчене звання)


(підпис)

І.Б. Туркін
(ініціали та прізвище)

1. Опис навчальної дисципліни

Найменування показника	Галузь знань, спеціальність, освітня програма, рівень вищої освіти	Характеристика навчальної дисципліни (денна форма навчання)
Кількість кредитів - 6	<p>Галузь знань <u>12 «Інформаційні технології»</u> <small>(шифр і найменування)</small></p> <p>Спеціальність <u>121 «Інженерія програмного забезпечення»</u> <small>(код і найменування)</small></p> <p>Освітня програма <u>«Інженерія програмного забезпечення»</u> <small>(найменування)</small></p> <p>Рівень вищої освіти: перший (бакалаврський)</p>	Цикл професійної підготовки, дисципліна вільного вибору студента
Кількість модулів - 2		Навчальний рік
Кількість змістовних модулів - 2		2019/2020
Індивідуальне завдання _____ <small>(назва)</small>		Семестр
		7-й
Загальна кількість годин – 72*/180		Лекції*
		40 годин
		Практичні, семінарські*
		_____ годин
		Лабораторні*
	32 години	
	Самостійна робота	
	108 годин	
	Вид контролю	
	модульний контроль, іспит	
Кількість тижневих годин для денної форми навчання: аудиторних – 4,5 самостійної роботи студента – 6,75		

Співвідношення кількості годин аудиторних занять до самостійної роботи становить: 72/108.

*Аудиторне навантаження може бути зменшене або збільшене на одну годину залежно від розкладу занять.

2. Мета та завдання навчальної дисципліни

Мета вивчення: надання студентам теоретичних знань і практичних навичок і прийомів в області проектування мережевих крос-платформних програмних додатків на широко розповсюдженій мові крос-платформного програмування Java, їхніх особливостях, архітектурі та перспективах розвитку.

Завдання: оволодіння студентами методів та технологій розробки крос-платформних програмних систем, оволодіння методами створення компонентів, дослідження архітектури та проектування компонентних програмних систем, розробка сценаріїв та алгоритмів взаємодії розподілених програмних додатків, ознайомлення з комунікаційними засобами розподілених обчислень.

Результати навчання:

У результаті вивчення даної навчальної дисципліни студент повинен:

Знати:

- основні вимоги, пропоновані до сучасних розподілених систем обробки інформації;
- основні принципи передачі даних у розподілених системах;
- методи та технології створення крос-платформних програмних систем;
- основи технології компонентного програмування та технології створення розподілених програм різної архітектури;
- сучасні моделі та технології взаємодії розподілених програмних систем обробки інформації;
- стеки мережних протоколів і інтерфейси;
- мережні служби та сервіси;

Вміти:

- самостійно застосовувати засоби крос-платформного програмування;
- самостійно розробляти прикладні протоколи та сценарії взаємодії мережевих додатків;
- самостійно розробляти та збирати компоненти програм;
- самостійно застосовувати проміжне програмне забезпечення;
- самостійно вибирати застосування, сервіси, компоненти та протоколи зв'язку
- проектувати та створювати компонентні мережеві програмні додатки на мові крос-платформного програмування Java.

Міждисциплінарні зв'язки: для успішного вивчення даної дисципліни необхідно мати базові знання з курсів: «Об'єктно-орієнтоване програмування», «Об'єктно-орієнтоване програмування на мові Java», «Компонентно-орієнтоване програмування», «Комп'ютерні мережі», «Операційні системи», «Бази даних», «Алгоритми та структури даних».

Знання та вміння, отримані студентом під час вивчення даної навчальної дисципліни, використовуються в подальшому при вивченні багатьох наступних дисциплін професійної підготовки фахівця з базовою та повною вищою освітою.

2. Програма навчальної дисципліни

Модуль 1

Змістовий модуль 1. Базові принципи побудови та взаємодії мережних крос-платформних додатків

ТЕМА 1. Предмет вивчення і задачі дисципліни «Крос-платформне програмування»

ТЕМА 2. Огляд сучасних технологій, які використовуються при створенні мережних крос-платформних програмних додатків

Задачі, для рішення яких використовуються розподілені системи. Основні вимоги, пропоновані до сучасних мережних розподілених систем обробки інформації. Узагальнена модель мережної розподіленої інформаційної системи. Огляд сучасних технологій реалізації розподілених систем: Java SE та Java EE Oracle, CORBA OMG, Microsoft .NET,. Перспективи розвитку розподілених систем та області їхнього застосування. Основні поняття архітектури розподілених мережних додатків: типові архітектури мережного клієнта і сервера. Типи мережних клієнтів та серверів.

ТЕМА 3. Взаємодія крос-платформних додатків з використанням потокових сокетів

Поняття мережного сокета: коротка історична довідка. Стек протоколів TCP/IP та модель ISO/OSI. Потокові та датаграмні сокети: основні особливості і області застосування. Організація взаємодії програмних додатків по протоколу TCP з використанням потокових сокетів. Алгоритми (сценарії) взаємодії клієнта і сервера при взаємодії через потокові сокети. Пакет засобів мережної підтримки мови Java java.net: призначення, архітектура, склад. Класи пакета java.net для реалізації потокових сокетів: InetAddress, ServerSocket, Socket та особливості їхнього застосування. Класи виключень, що виникають при роботі з потоковими сокетами.

ТЕМА 4. Взаємодія крос-платформних додатків з використанням датаграмних сокетів

Організація взаємодії розподілених додатків по протоколу UDP з використанням датаграмних сокетів. Алгоритми (сценарії) взаємодії клієнта і сервера при взаємодії через датаграмні сокети. Класи пакета java.net для реалізації датаграмних сокетів: DatagramPacket, DatagramSocket та особливості їхнього застосування. Класи виключень, що виникають при роботі з датаграмними сокетами.

ТЕМА 5. Взаємодія мережних додатків за допомогою URL

Уніфіковані ідентифікатори, покажчики та імена ресурсів мережі Internet (URI, URL і URN) – призначення, синтаксис. Типові схеми URL. Класи та інтерфейси пакета java.net для реалізації взаємодії на рівні URL: java.net.URL, java.net.URLConnection і особливості їхнього застосування. Оброблювачі протоколу та вмісту.

ТЕМА 6. *Взаємодія мережних додатків з використанням протоколу HTTP*

Протокол HTTP: призначення, область застосування, версії, коротка історична довідка. Типи MIME – призначення, види, застосування. Структура запитів та відповідей протоколу HTTP. Команди і поля заголовків протоколу HTTP. Коди стану протоколу HTTP. Алгоритм (сценарій) взаємодії мережних розподілених додатків по протоколу HTTP.

ТЕМА 7. *Взаємодія мережних додатків з використанням інтерфейсу CGI*

Інтерфейс CGI: призначення, архітектура, області застосування, достоїнства та недоліки. Технології альтернативні CGI. Змінні оточення CGI. Принципи використання інтерфейсу CGI у мережних крос-платформних додатках: алгоритм (сценарій) взаємодії мережних додатків з використанням сценаріїв CGI. Робота з CGI-сценаріями в додатках мовою Java.

ТЕМА 8. *Взаємодія мережних додатків на рівні викликів методів вилучених об'єктів*

Технологія Remote Method Invocation (RMI): коротка історична довідка. RMI та RPC (Remote Procedure Call) – загальні риси і розходження. Области застосування технології RMI. Архітектура RMI: вилучені об'єкти та вилучені інтерфейси, стаби і каркаси, служба імен (реєстр об'єктів). Механізми взаємодії вилучених об'єктів: серіалізація об'єктів, згортання-розгортання параметрів при викликах вилучених методів і поверненні результату через мережу. Загальна модель (сценарій) взаємодії мережних додатків при використанні технології RMI. Протоколи RMI/JRMP та RMI/IIOP: призначення, застосування, особливості. Передача викликів RMI через протокол HTTP.

ТЕМА 9. *Реалізація технології RMI засобами мови Java: технологія Java RMI*

Технологія Java RMI та протокол RMI/JRMP – призначення, основні можливості, області застосування. Архітектура Java RMI: основні пакети класів та інтерфейсів; складові частини Java RMI: ядро RMI, служба імен (реєстр), система безпеки, служби нижнього рівня. Діаграми основних класів та інтерфейсів. Передача параметрів при викликанні методів вилучених об'єктів: основні правила і особливості. Зворотний виклик. Загальна методика проектування мережних додатків на основі технології Java RMI/JRMP.

ТЕМА 10. *Основи безпеки мережних Java-додатків*

Загальні відомості про життєздатність та безпеку мережних систем обробки інформації. Базова модель захисту. Моделі систем безпеки Java-додатків: “sandbox”, “trusted” та доменна, їхні характерні риси, переваги та недоліки. Архітектура системи захисту платформи Java 2. Ядро системи безпеки Java: верифікатор байт-коду, завантажники класів, диспетчер захисту та контролер доступу; призначення, основні функції. Керування доступом: домени безпеки, права доступу, класи прав доступу, політики безпеки та адміністрування. Додаткові служби безпеки: Java Authentication and Authorization Service (JAAS), Java Secure Socket Extension (JSSE) та Java Cryptography Extension (JCE) – призначення, функції.

Модуль 2

Змістовий модуль 2. Організація взаємодії та доступу до мережесих ресурсів у крос-платформених додатках

ТЕМА 1. Доступ до ресурсів у розподілених системах. Інтерфейс Java Naming and Directory Interface (JNDI)

Загальні зведення про служби імен та каталогів: призначення, області застосування, основні визначення, приклади існуючих служб імен і каталогів (NIS, NDS, LDAP, DNS, Naming Service CORBA, File Systems) та їхні характерні риси. Взаємодія Java-додатків з службами імен та каталогів. Java JNDI API: призначення, область застосування, основні переваги, архітектура, пакети. Основні інтерфейси та класи JNDI API. Особливості настроювання JNDI API.

ТЕМА 2. Взаємодія мережесих крос-платформених додатків з використанням технології Java RMI/IIOP

Технологія Java RMI та протокол IIOP – призначення, основні можливості, області застосування. Основні пакети класів та інтерфейсів підтримки взаємодії розподілених крос-платформених додатків за допомогою технології Java RMI/IIOP. Загальна методика проектування крос-платформених додатків на основі технології Java RMI/IIOP.

ТЕМА 3. Взаємодія мережесих крос-платформених додатків з використанням технології CORBA

Технологія CORBA (Common Object Request Broker Architecture): призначення, область застосування, версії, коротка історична довідка. Архітектура CORBA: компонентна модель OMG-OMA, сервіси CORBA. Брокери об'єктних запитів (Object Request Broker): призначення та архітектура. Сервіс імен CORBA. Протоколи CORBA. Мова опису інтефейсів IDL (Interface Definition Language). Сценарії взаємодії крос-платформених мережесих додатків при використанні технології CORBA. Реалізація CORBA на платформі Java: технологія Java IDL. Загальна методика проектування крос-платформених додатків на основі технології Java IDL.

ТЕМА 4. Взаємодія мережесих крос-платформених додатків з базами даних. Технологія Java JDBC

Технологія JDBC: призначення, основні можливості, область застосування, переваги, етапи розвитку. Архітектура JDBC: принципи побудови JDBC, основні класи та інтерфейси. Драйвери та з'єднання в JDBC: класифікація драйверів по функціональності і засобам реалізації; завантаження драйверів та встановлення з'єднання з базами даних. Доступ до баз даних за допомогою JNDI: інтерфейс javax.sql.DataSource. Джерела даних. Пули з'єднань. Завантаження класу драйвера. З'єднання з базою даних. Використання файлу властивостей. Робота з даними: одержання та відновлення даних, пакетні зміни, відповідність між типами SQL і Java. Результуючі набори даних: навігація по

набору даних, модифікація набору даних. Обробка виняткових ситуацій. Робота зі збереженими процедурами баз даних. Метадані: одержання інформації про базу даних у цілому, одержання інформації про таблиці бази даних.

ТЕМА 5. *Проектування крос-платформних мережевих додатків на базі платформи Java EE*

Платформа Java EE (Java Enterprise Edition): призначення. Короткий огляд Java EE: компонентна архітектура, архітектура MVC, API, специфікація, переваги Java EE. Технології платформи Java EE: компонентні технології, технології служб, комунікаційні технології. Типи контейнерів і компонентів платформи Java EE.

Проектування та розробка Java EE-додатків: рівні Java EE -моделі (клієнтський рівень, рівень представлення даних, рівень бізнес-логіки, рівень даних); компоненти Java EE-додатків (компоненти клієнта. Web-компоненти, бізнес-компоненти). Основні бізнес-сценарії, описувані за допомогою Java-компонентів. Сервери додатків. Структура серверних додатків. Збірка та розгортання додатків на сервері додатків. Структура клієнтських додатків Java EE. Ролі, передбачені платформою Java EE.

ТЕМА 6. *Проектування керуючої логіки Web- додатка. Технологія Java Servlet API*

Технологія Java Servlets: призначення, основні можливості, область застосування, переваги перед CGI, етапи розвитку. Архітектура сервлетів: пакети Java Servlet API - основні класи та інтерфейси, життєвий цикл сервлета. Прийоми роботи із сервлетами: створення та запуск сервлета, доступ до сервлету, обробка запитів, передача параметрів, генерація відповідей, керування сеансом (збереження стану), взаємодія з базами даних (JDBC), взаємодія між сервлетами (перенапрямок запитів), конфігурація сервлетів.

ТЕМА 7. *Проектування логіки представлення даних. Технологія Java Server Pages*

Технологія Java Server Pages (JSP): призначення, основні можливості, область застосування, переваги. Архітектура JSP: основні моделі архітектури JSP (JSP Model 1 та JSP Model 2), життєвий цикл JSP. Синтаксис JSP-сторінки: коментарі, теги, елементи-директиви, елементи-дії, елементи сценарію. Область видимості об'єктів, неявні об'єкти. Обробка помилок. Розширення функціональності JSP-сторінок: клас оброблювача тега, бібліотеки тегів, оголошення бібліотек тегів. Використання компонентів JavaBeans в JSP-сторінках. Взаємодія JSP-сторінок та сервлетів.

Висновки. Перспективи розвитку сучасних мережевих крос-платформних систем обробки інформації.

4. Структура навчальної дисципліни

Назва змістовного модуля і тем	Кількість годин				
	Усього	У тому числі			
		л	п	лаб.	с. р.
1	2	3	4	5	6
Модуль 1					
Змістовний модуль 1. Базові принципи побудови та взаємодії мережних крос-платформних додатків					
Тема 1. Предмет вивчення і задачі дисципліни «Крос-платформне програмування»	1	1	-	-	-
Тема 2. Огляд сучасних технологій, які використовуються при створенні мережесих крос-платформних програмних додатків	6	2	-	-	4
Тема 3. Взаємодія крос-платформних додатків з використанням потокових сокетів	12	2	-	4	6
Тема 4. Взаємодія крос-платформних додатків з використанням датаграмних сокетів	12	2	-	4	6
Тема 5. Взаємодія мережесих додатків за допомогою URL	10	2	-	4	4
Тема 6. Взаємодія мережесих додатків з використанням протоколу HTTP	6	2	-	-	4
Тема 7. Взаємодія мережесих додатків з використанням інтерфейсу CGI	5	1	-	-	4
Тема 8. Взаємодія мережесих додатків на рівні викликів методів вилучених об'єктів	6	2	-	-	4
Тема 9. Реалізація технології RMI засобами мови Java: технологія Java RMI	10	2	-	4	4
Тема 10. Основи безпеки мережесих Java-додатків	6	2	-	-	4
Модульний контроль	8	-	-	-	8
Разом за змістовним модулем 1	82	18	-	16	48
Усього годин	82	18	-	16	48
Модуль 2					

Змістовний модуль 2. Організація взаємодії та доступу до мережесвих ресурсів у крос-платформених додатках					
Тема 1. Доступ до ресурсів у розподілених системах. Інтерфейс Java Naming and Directory Interface (JNDI)	6	2	-	-	4
Тема 2. Взаємодія мережесвих крос-платформених додатків з використанням технології Java RMI/IIOP	12	2	-	4	6
Тема 3. Взаємодія мережесвих крос-платформених додатків з використанням технології CORBA	14	4	-	4	6
Тема 4. Взаємодія мережесвих крос-платформених додатків з базами даних. Технологія Java JDBC	14	4	-	4	6
Тема 5. Проектування крос-платформених мережесвих додатків на базі платформи Java EE	12	4	-	-	8
Тема 6. Проектування керуючої логіки Web- додатка. Технологія Java Servlet API	10	2	-	2	6
Тема 7. Проектування логіки представлення даних. Технологія Java Server Pages	12	4	-	2	6
Модульний контроль	8	-	-	-	8
Разом за змістовним модулем 2	88	22	-	16	50
Усього годин	88	22	-	16	50
Індивідуальне завдання	-	-	-	-	-
Контрольний захід	10	-	-	-	10
Усього годин	180	40	-	32	108

5. Теми семінарських занять

№ п/п	Назва теми	Кількість годин
1		
2		
	Разом	

6. Теми практичних занять

№ п/п	Назва теми	Кількість годин
1		
2		
	Разом	

7. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Взаємодія крос-платформених додатків з використанням потокових сокетів TCP/IP	4
2	Взаємодія крос-платформених додатків з використанням дейтаграмних сокетів UDP/IP	4
3	Взаємодія мережних розподілених додатків на рівні URL-ресурсів	4
4	Взаємодія крос-платформених додатків з використанням технології Java RMI/JRMP	4
5	Взаємодія крос-платформених додатків з використанням технології Java RMI/IIOP	4
6	Взаємодія крос-платформених додатків з використанням технології CORBA (Java IDL)	4
7	Взаємодія крос-платформених додатків з базами даних з використанням технології Java JDBC	4
8	Взаємодія крос-платформених додатків з використанням технологій Java Servlets та JSP	4
	Разом	32

8. Самостійна робота

№ з/п	Назва теми	Кількість годин
1	Огляд сучасних технологій, які використовуються при створенні крос-платформених додатків	4
2	Взаємодія мережних крос-платформених додатків на рівні потокових сокетів	6
3	Взаємодія мережних крос-платформених додатків на рівні дейтаграмних сокетів	6
4	Взаємодія мережних крос-платформених додатків за допомогою URL	4
5	Взаємодія мережних крос-платформених додатків з використанням протоколу HTTP	4
6	Взаємодія мережних крос-платформених додатків з використанням інтерфейсу CGI	4
7	Взаємодія мережних крос-платформених додатків на рівні викликів методів вилучених об'єктів	4
8	Реалізація технології RMI засобами мови Java: технологія Java RMI	4

9	Основи безпеки мережевих Java-додатків	4
10	Доступ до ресурсів у розподілених системах. Інтерфейс Java Naming and Directory Interface (JNDI)	4
11	Взаємодія мережних крос-платформних додатків з використанням технології Java RMI/IIOP	6
12	Взаємодія мережних крос-платформних додатків з використанням технології CORBA	6
13	Взаємодія мережних крос-платформних додатків з базами даних. Технологія Java JDBC	6
14	Пректування крос-платформних мережевих додатків на базі платформи Java EE	8
15	Проектування керуючої логіки Web- додатка. Технологія Java Servlet API	6
17	Проектування логіки представлення даних. Технологія Java Server Pages	6
18	Підготовка до контрольних заходів	26
	Разом	108

9. Індивідуальні завдання

Не передбачено навчальним планом.

10. Методи навчання

1. За джерелами придбання знань – словесні: лекція (вступна, традиційна, проблемна, з помилками), бесіда (евристична), диспут, дискусія, робота з друкованими та інтернет-джерелами; наочні: ілюстрація, спостереження; практичні: вправа, лабораторна робота.
2. За характером пізнавальної діяльності тих, хто навчається – інформаційно-репродуктивний, репродуктивний, проблемне викладання, частково-пошуковий.
3. За логікою пізнання – індуктивний, дедуктивний, аналогій, вивідних знань.
4. Методи перевірки й оцінки знань, умінь, навичок: спостереження, усне опитування, контрольні роботи, програмований контроль, тестування (традиційне та машинне).

11. Методи контролю

1. Опитування.
2. Лабораторні роботи.
3. Модульні контрольні роботи.

Форма підсумкового контролю успішності навчання: іспит (письмово) у 7 семестрі.

12. Критерії оцінювання та розподіл балів, які отримують студенти

12.1. Розподіл балів, які отримують студенти (кількісні критерії оцінювання)

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Змістовний модуль 1			
Робота на лекціях	0...1	10	0...10
Виконання і захист лабораторних (практичних) робіт	3...5	4	12...20
Модульний контроль	0...20	1	0...20
Змістовний модуль 2			
Робота на лекціях	0...1	10	0...10
Виконання і захист лабораторних (практичних) робіт	3...5	4	12...20
Модульний контроль	0...20	1	0...20
Виконання і захист РГР (РР, РК)	-	-	-
Усього за семестр			60...100

Семестровий контроль (іспит) проводиться у разі відмови студента від балів поточного тестування й за наявності допуску до іспиту/заліку. Під час складання семестрового іспиту/заліку студент має можливість отримати максимум 100 балів.

Білет для іспиту/заліку складається з трьох теоретичних питань. Відповідь на перше питання оцінюється від 0 до 34 балів, відповідь на друге питання оцінюється від 0 до 33 балів, відповідь на третє питання оцінюється від 0 до 33 балів.

12.2. Якісні критерії оцінювання

Необхідний обсяг знань для одержання позитивної оцінки: основні вимоги до мережевих крос-платформних додатків, моделі архітектури та взаємодії програмних додатків, технології, які використовуються для розробки та взаємодії крос-платформених мережевих програмних додатків та особливості їх застосування, їх переваги і недоліки, мережеві протоколи та алгоритми (сценарії) взаємодії програмних додатків, основи забезпечення безпеки мережевих додатків, засоби і технології взаємодії крос-платформених

додатків з службами імен і каталогів, базами даних та мережевими інформаційними ресурсами.

Необхідний обсяг вмінь для одержання позитивної оцінки: обґрунтовано робити вибір моделі та технології взаємодії крос-платформених додатків залежно від складності вирішуваних завдань. та використовувати її при розробці програмного забезпечення.

12.3 Критерії оцінювання роботи студента протягом семестру

Задовільно (60-74). Мати мінімум знань та умінь. Відпрацювати та захистити всі лабораторні роботи та виконати контрольні завдання передбачені у модульному контролі. Вміти використовувати технології взаємодії крос-платформених додатків при розробці програмного забезпечення.

Добре (75 - 89). Твердо знати мінімум знань, виконати усі завдання. Показати вміння виконувати та захищати всі лабораторні роботи в обумовлений викладачем строк з обґрунтуванням рішень та заходів, які запропоновано у роботах. Вміти пояснювати вибір технології взаємодії та розробки крос-платформених мережових додатків.

Відмінно (90 - 100). Повно знати основній та додатковий матеріал. Знати усі теми. Орієнтуватися у підручниках та посібниках. Досконально знати усі технології, які використовуються при проектуванні крос-платформених мережових додатків. Вміти обґрунтовано робити вибір технології взаємодії крос-платформених додатків залежно від складності вирішуваних завдань. та використовувати її при розробці програмного забезпечення.. Безпомилково виконувати та захищати всі лабораторні роботи в обумовлений викладачем строк з докладним обґрунтуванням рішень та заходів, які запропоновано у роботах.

Шкала оцінювання: бальна і традиційна

Сума балів	Оцінка за традиційною шкалою	
	Іспит, диференційований залік	Залік
90 – 100	Відмінно	Зараховано
75 – 89	Добре	
60 – 74	Задовільно	
0 – 59	Незадовільно	Не зараховано

13. Методичне забезпечення

1. Розроблений лекційний курс та комплекс презентацій Power Point ([//master/студенты/2019-2020/6 факультет/4курс/Крос-платф. програмування Матеріали курсу/Лекции](#)).
2. Розроблені питання для модульних контрольних робіт ([//master/студенты/2019-2020/6 факультет/4курс/Крос-платф. програмування /Модули](#)).

3. Розроблені питання для підсумкового контролю успішності навчання ([//master/студенты/2019-2020/6 факультет/4курс/Крос-платф. програмування /Вопросы](#)).
4. Лабораторні роботи та методичний посібник ([//master/студенты/2019-2020/6 факультет/4курс/Крос-платф. програмування /Лабораторные работы](#)).
5. Дібрані матеріали для самостійної роботи студентів ([//master/студенты/2019-2020/6 факультет/4курс/Крос-платф. програмування /Books](#)).

14. Рекомендована література

Базова

1. Гослинг Д., Джой Б., Стил Г., Гилад Брача Г., Бакли А.. Язык программирования Java SE 8. Подробное описание, 5-е издание = The Java Language Specification, Java SE 8 Edition (5th Edition) (Java Series). — М.: «Вильямс», 2015. — 672 с.
2. Дилип Н. Стандарты и протоколы Интернета. /Пер. с англ.- М.: Издательский отдел «Русская редакция» ТОО «Channel Trading Ltd.», 1999. - 384 с.
3. Герберт Шилдт. Java 8. Полное руководство, 9-е издание = Java 8. The Complete Reference, 9th Edition. — М.: «Вильямс», 2015. — 1376 с
4. Васильев А. Н. Java. Объектно-ориентированное программирование. - СПб.: «Питер», 2012. - 398 с.
5. Нимейер П., Леук Д. Программирование на Java. /Пер. с англ.— М.: «ЭКСМО», 2014. - 1216 с.
6. Гонсалвес Э. Изучаем Java EE 7 /Пер. с англ.— СПб.: «Питер», 2014. - 640 с.
7. Гери Д., Хорстман К. JavaServer Faces. Библиотека профессионала. 6-е изд. /Пер. с англ.— М.: ООО «И. Д. Вильямс», 2011. - 544 с.
8. Перроун П., Чаганти В. Создание корпоративных систем на основе Java 2 Enterprise Edition. Руководство разработчика. /Пер. с англ. - М.: Издательский дом «Вильямс», 2001. - 1184 с.
9. Хорстманн К., Корнелл Г. Библиотека профессионала. Java 2. Том 1, 2. /Пер. с англ. - М.: Издательский дом «Вильямс», 2002. - 1120 с.
10. Цимбал А., Аншина М. Технологии создания распределенных систем. Для профессионалов. / СПб.: «Питер», 2003. – 576 с.
11. Риккарди Г. Системы баз данных. Теория и практика использования в Internet и среде Java. /Пер. с англ. - М.: Издательский дом «Вильямс», 2001. - 480 с.
12. Хабибуллин И. Создание распределенных приложений на Java 2. - СПб.: «БХВ – Петербург», 2002. - 704 с.

Допоміжна

1. Буч Г., Рамбо Д., Джекобсон А. Язык UML. Руководство пользователя. /Пер. с англ. – М.: «ДМК», 2000. – 432 с.

2. Ларман К. Применение UML и шаблонов проектирования. Практическое руководство. 3-е издание. /Пер. с англ. - М.: ООО «И. Д. Вильямс», 2009. - 736 с.
3. Перроун П., Яворски Д. Система безопасности Java. Руководство разработчика. /Пер. с англ.- М.: Издательский дом «Вильямс», 2001.- 528 с.

15. Інформаційні ресурси

1. Java SDK, Standard Edition Documentation:
<http://www.oracle.com/technetwork/java/javase/overview/index.html>
2. The Java Tutorials. A practical guide for programmers:
<http://docs.oracle.com/javase/tutorial/>
3. Java Resources for Developers:
<http://www.oracle.com/technetwork/java/index.html>
4. The Java EE 7 Tutorial:
<http://docs.oracle.com/javaee/7/tutorial/doc/home.htm>
5. About the Common Object Request Broker Architecture Specification:
<https://www.omg.org/spec/CORBA/About-CORBA/>
6. W3C standards:
<https://www.w3.org/standards/>