Затверджено Вченою Радою ОНУ
імені І.І. Мечникова
від "20"грудня 20 16 р. № 4

Одеський національний університет імені І.І. Мечникова
(повна назва вищого навчального закладу)

Кафедра комп'ютерної алгебри та дискретної математики

"**ЗАТВЕРДЖУЮ**"
Проректор з науково-педагогічної роботи
_____ (П.І.Б.)
"____" _____ 20___ р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Багатониткове програмування в Java
(назва навчальної дисципліни)

Рівень вищої освіти _____

Спеціальність 123 – Комп'ютерна інженерія
(код і назва спеціальності (тей)

Інститут/факультет факультет математики, фізики та інформаційних технологій
(назва інституту, факультету)

2020 – 2021

Робоча програма складена на основі навчальної програми з дисципліни «Багатониткове програмування в Java» .
(назва навчальної дисципліни)

Розробники: Варбанець С.П., к. ф.-м. н., доцент.

Робоча програма затверджена на засіданні кафедри комп. алг. та дискр. мат.

Протокол № 1 від. "31" серпня 2020 р.

Завідувач кафедри _____ (підпис) (Варбанець П.Д.) (прізвище та ініціали)

Схвалено навчально-методичною комісією (НМК)_____ факультету/інституту

Протокол № 1 від. "31" 08 2020 р.

Голова НМК _____ (підпис) (Савастру) (прізвище та ініціали)

Переглянуто та затверджено на засіданні кафедри _____

Протокол № ___ від. "____"_____20__ р.

Завідувач кафедри _____ (підпис) (_____) (прізвище та ініціали)

Переглянуто та затверджено на засіданні кафедри _____

Протокол № ___ від. "____"_____20__ р.

Завідувач кафедри _____ (підпис) (_____) (прізвище та ініціали)

1. Опис навчальної дисципліни

<table>
<tr><td rowspan="2">Найменування показників</td><td rowspan="2">Галузь знань, спеціальність, спеціалізація, рівень вищої освіти</td><td colspan="2">Характеристика навчальної дисципліни</td></tr>
<tr><td>денна форма навчання</td><td>заочна форма навчання</td></tr>
<tr><td rowspan="15">Загальна кількість: кредитів –4

годин – 120

залікових модулів – 1

змістових модулів – 2

ІНДЗ – _____
(вид завдання)</td><td rowspan="15">Галузь знань
12 Інформаційні технології
(шифр і назва)

Спеціальність
123 – Комп’ютерна інженерія
(код і назва)

Спеціалізації:
_____
(назва)

Рівень вищої освіти:
магістри</td><td colspan="2">Нормативна / за вибором (ВНЗ/студента)</td></tr>
<tr><td colspan="2">Рік підготовки:</td></tr>
<tr><td>2020</td><td>–</td></tr>
<tr><td colspan="2">Семестр</td></tr>
<tr><td>1</td><td>–</td></tr>
<tr><td colspan="2">Лекції</td></tr>
<tr><td>18</td><td>–</td></tr>
<tr><td colspan="2">Практичні, семінарські</td></tr>
<tr><td>16</td><td>–</td></tr>
<tr><td colspan="2">Лабораторні</td></tr>
<tr><td>–</td><td>–</td></tr>
<tr><td colspan="2">Самостійна робота</td></tr>
<tr><td>86</td><td>–</td></tr>
<tr><td colspan="2">у т.ч. ІНДЗ: –</td></tr>
<tr><td colspan="2">Форма підсумкового контролю:
іспит</td></tr>
</table>

2. Мета та завдання навчальної дисципліни

Мета курсу – ознайомити студентів з сучасними засобами багатониткового програмування на прикладі об'єктно-орієнтованої мови програмування Java. Навчити студентів користуватися отриманими теоретичними знаннями в розв'язанні проблем, які виникають в практичних задачах реалізації паралельного розгалуження алгоритмів виконання програм.

Завдання полягає у розгляді всіх аспектів багатониткового програмування та в формуванні у студентів паралельного мислення у функціонуванні програм.

Процес вивчення дисципліни спрямований на формування елементів наступних **компетентностей**:

а) загальних (ЗК): КІ.01, КІ.02, КІ.03

б) фахових загальних (КФЗ): КЗП.01, КЗП.02, КЗП.03

в) спеціальних фахових (КФС): КСП.01, КСП.05, КСП.06

Очікувані результати навчання. У результаті вивчення навчальної дисципліни студент повинен

знати: основні аспекти багатониткового програмування.

вміти: обирати оптимальний план реалізації паралелізму в залежності від наявної проблеми взаємодії між кількома окремими компонентами інформаційної системи.

3. Зміст навчальної дисципліни

Змістовий модуль 1. Традиційна модель багатонитковості в Java

Тема 1. Модель ниток виконання в Java. Головна нитка виконання.

Тема 2. Створення декількох ниток виконання. Пріоритети ниток виконання.

Тема 3. Традиційні методи синхронізації. Взаємодія ниток виконання.

Тема 4. Призупинення, відновлення та зупинка ниток виконання. Стан нитки виконання. Одночасне створення і запуск ниток виконання фабричними методами.

Змістовий модуль 2. Службові засоби паралелізму

Тема 1. Пакети паралельного прикладного програмного інтерфейсу API.

Тема 2. Синхронізатори. Семафори. Самоблокування із зворотнім відліком. Використання циклічних бар'єрних точок. Обмін даними між нитками. Синхронізація ниток стадії виконання дій.

Тема 3. Виконувачі. Класи і інтерфейси виконувачів. Клас Executor. Інтерфейси Future і Callable.

Тема 4. Паралельні колекції.

Тема 5. Поняття блокувань.

Тема 6. Атомарні операції паралелізму.

Тема 7. Каркас Fork/Join Framework. Ядро каркасу. Стратегія «розділяй і володій». Клас RecursiveTask. Асинхронне виконання задач. Визначення стану задачі. Класи ForkJoinTask і ForkJoinPool.

4. Структура навчальної дисципліни

Назви тем	Кількість годин									
	Денна форма					Заочна форма				
	Усього	у тому числі				Усього	у тому числі			
		л	п/с	лаб	ср		л	п/с	лаб	ср
1	2	3	4	5	6	7	8	9	10	11
Змістовий модуль 1. Традиційна модель багатонитковості в Java.										
Тема 1. Модель ниток виконання в Java. Головна нитка виконання.	2	1	1	0	8					
Тема 2. Створення декількох ниток виконання. Пріоритети ниток виконання.	2	1	1	0	8					
Тема 3. Традиційні методи синхронізації. Взаємодія ниток виконання.	3	2	1	0	6					
Тема 4. Призупинення, відновлення та зупинка ниток виконання. Стан нитки виконання. Одночасне створення і запуск ниток виконання фабричними методами.	3	2	1	0	12					
Разом за змістовим модулем 1	10	6	4	0	34					
Змістовий модуль 2. Службові засоби паралелізму.										
Тема 1. Пакети	2	2	0	0	10					

паралельного прикладного програмного інтерфейсу API.										
Тема 2. Синхронізатори. Семафори. Самоблокування із зворотнім відліком. Використання циклічних бар'єрних точок. Обмін даними між нитками. Синхронізація ниток стадії виконання дій.	6	2	4	0	10					
Тема 3. Виконувачі. Класи і інтерфейси виконувачів. Клас Executor. Інтерфейси Future і Callable.	4	2	2	0	10					
Тема 4. Паралельні колекції.	4	2	2	0	4					
Тема 5. Поняття блокувань.	3	1	2	0	4					
Тема 6. Атомарні операції паралелізму.	1	1	0	0	9					
Тема 7. Каркас Fork/Join Framework. Ядро каркасу. Стратегія «розділяй і володій». Клас RecursiveTask. Асинхронне виконання задач. Визначення стану задачі. Класи ForkJoinTask і ForkJoinPool.	4	2	2	0	9					

Разом за змістовим модулем 2	24	12	12		52					
ІНДЗ										
Усього годин	34	18	16		86					

5. Теми практичних занять

№ з/п	Назва теми	Кількість годин
1	Головна нитка. Створення декількох ниток виконання. Визначення ниток за іменем. Пріоритети ниток.	2
2	Синхронізація та взаємодія ниток виконання. Призупинення, відновлення та зупинка нитки. Визначення стану нитки.	2
3	Семафори. Самоблокування із зворотнім відліком.	2
4	Циклічні бар'єрні точки. Обмін даними між нитками. Синхронізація ниток стадії виконання дій.	2
5	Клас Executor. Інтерфейси Future і Callable.	2
6	Паралельні колекції.	2
7	Методи блокування ниток.	2
8	Реалізація стратегії «розділяй і володій». Клас RecursiveTask. Асинхронне виконання задач.	2

6. Методи навчання

Курс поділено на два етапи. Перший етап (теоретична частина) передбачає аудиторні заняття у вигляді презентаційних лекцій з використанням мультимедійного пристрою демонстрації слайдів. Другий етап (практична частина) полягає у виконанні покрокових практичних завдань в комп'ютерному класі з подальшим виконанням індивідуальних завдань, відповідно до тем практичних завдань. Бали, отримані за окремі індивідуальні завдання формують загальні бали відповідних змістовних модулів.

7. Методи контролю

Метод контролю передбачає проведення двох модульних контрольних робіт відповідно до тем змістовних модулів.

8. Питання для підсумкового контролю

Підсумковий контроль складається з проведення іспиту по наступних питаннях:

1. Як ви розумієте багатонитковість? Опишіть модель ниток виконання в Java.
2. За яким принципом виконується перемикання контексту?
3. Опишіть принцип синхронізації ниток. Що таке монітор?
4. Опишіть принцип обміну повідомленнями в багатонитковій системі.
5. У чому полягає різниця між Thread і Runnable? Яким чином виконується нитка? Як її запустити?

6. Що таке головна нитка? Як її створити? Як відрізнити головну нитку від інших?
7. Які ви знаєте способи створення ниток? Будь ласка, опишіть кожен з них.
8. Як можна визначити, чи завершила певна нитка свою роботу? Опишіть на прикладі.
9. Як змусити певну нитку дочекатись завершення роботи декількох, або всіх інших ниток? Опишіть на прикладі.
10. Як визначити або встановити пріоритет певної нитки?
11. Опишіть як відбувається синхронізація методами класу. Наведіть приклад.
12. Опишіть метод блокової синхронізації. В чому її перевага перед синхронізації методом.?
13. Опишіть процес блокування та розблокування ниток. Дайте характеристику відповідним методам. В чому полягає небезпека взаємного блокування?
14. Які ви знаєте способи призупинки, поновлення та зупинки ниток? Поясніть, в чому полягають їх недоліки.
15. Опишіть життєвий цикл окремої нитки. В яких станах може знаходитись нитка і як це відслідити?
16. Що таке семафор? Дайте характеристику та опишіть алгоритм використання семафорів.
17. В чому полягає призначення самоблокування із зворотнім відліком? Переваги та недоліки.
18. Що таке циклічний бар'єр і як його використовувати?
19. Як реалізувати обмін даними між двома нитками?
20. Синхронізація методом фаз. Поясніть принцип її роботи.
21. В чому полягає головне призначення виконавців? Опишіть які ви знаєте класи виконавців та принципи їх роботи.
22. Інтерфейси Future і Callable. Опишіть їх призначення, переваги та недоліки.
23. Які ви знаєте паралельні колекції?
24. Блокування ниток за допомогою пакету java.util.concurrent.locks. Які методи блокування вам відомі?
25. Що таке атомарність? Назвіть переваги та недоліки атомарних операцій.
26. Опишіть принцип роботи класу ForkJoinTask.
27. Опишіть принцип роботи класу ForkJoinPool.
28. Опишіть стратегію «розділяй та володій». Класи RecursiveAction і RecursiveTask.
29. Як можна реалізувати асинхронне виконання задач?
30. Що таке реактивна нитка? Засоби контролю ниток даних.

9. Розподіл балів, які отримують студенти

Орієнтовний приклад для екзамену

Поточний контроль		Модульний контроль 1	Модульний контроль 2	Підсумковий контроль	Сума балів
Змістовий модуль 1	Змістовий модуль 2				
10	30	20	30	10	100

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90 – 100	**A**	відмінно	зараховано
85-89	**B**	добре	
75-84	**C**		
70-74	**D**	задовільно	
60-69	**E**		
35-59	**FX**	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	**F**	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

10. Рекомендована література

Основна

1. Герберт Шилдт, Java. Полное руководство, 10-е изд., 2018.
2. Кей Хорстман, Java. Библиотека профессионала. Том 1. Основы, 11-е изд., 2019.

Додаткова

1. Герберт Шилдт, Java. Руководство для начинающих, 7-е изд., 2019.
2. Джошуа Блох, Java. Эффективное программирование, 3-е изд., 2019.
3. Кей Хорстман, Java. Библиотека профессионала. Том 1. Основы, 10-е изд., 2016.
4. Брайан Гетц, Тим Пайерлс, Джошуа Блох, Джозеф Боубер, Дэвид Холмс, Даг Ли, Java Concurrency на практике, 2020.

11. Електронні інформаційні ресурси

1. https://docs.oracle.com/javase/tutorial/essential/concurrency/index.html