

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА

ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ

Кафедра математичної інформатики



«ЗАТВЕРДЖУЮ»

Заступник декана
з навчальної роботи

Кашпур О.Ф.

« 28 » серпня 2020 року

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
СКЛАДНІСТЬ АЛГОРИТМІВ**

для студентів

галузь знань **12 «Інформаційні технології»**
(шифр і назва)

спеціальність **122 «Комп'ютерні науки»**
(шифр і назва спеціальності)

освітній рівень **бакалавр**
(молодший бакалавр, бакалавр, магістр)

освітня програма **«Інформатика»**
(назва освітньої програми)

вид дисципліни **вибіркова**

вибірковий блок **«Інтелектуальні інформаційні технології»**

Форма навчання **заочна**

Навчальний рік **2020/2021**

Семестр **7**

Кількість кредитів ECTS **3**

Мова викладання, навчання та оцінювання **українська**

Форма заключного контролю **залік**

Пролонговано: на 2021/2022 н. р.


на 20__/20__ н. р.



« 7 » 05 2021 р.

) « » 20__ р.

Розробник:

Федорус Олексій Мстиславович, асистент, кафедра математичної інформатики 


Терещенко Василь Миколайович, д.ф.-м.н., професор, завідувач кафедри математичної інформатики

ЗАТВЕРДЖЕНО

Завідувач кафедри математичної інформатики

 Терещенко В.М.

Протокол № 1 від «28» 08 2020 р.

Схвалено  Гарантом освітньо-професійної програми «Інформатика»
(Омельчук Л.Л.)

« 28 » 08 2020 р.

Схвалено науково-методичною комісією факультету комп'ютерних наук та кібернетики

Протокол від « 28 » 08 2020 року № 1

Голова науково-методичної комісії  (Омельчук Л.Л.)

« 28 » 08 2020 року

1. Мета дисципліни "Аналіз складності алгоритмів" є набуття теоретичних та практичних знань в одній з найактуальніших на сьогодні галузей інформаційних технологій: аналіз алгоритмів та визначення їх складності. У ході навчання студенти ознайомляться з основними моделями для аналізу алгоритмів та визначенню їх складності.

2. Попередні вимоги до опанування або вибору навчальної дисципліни:

1. *Успішне опанування курсів «Програмування».*

2. *Володіти сучасними програмними мовами, засобами розробки та проектування програм для розв'язання наукових та прикладних задач та навичками навчальної діяльності.*

3. Анотація навчальної дисципліни:

Навчальна дисципліна «Командна розробка програмного продукту» є складовою освітньо-професійної програми підготовки фахівців за першим (бакалавр) рівнем вищої освіти *галузі знань 12 „Інформаційні технології” зі спеціальності 122 „Комп’ютерні науки”, освітньо-професійної програми „Інформатика”.*

Дана дисципліна є дисципліною за вибором для спеціальності **“Інформатика”**.

Викладається у **2 семестрі 4 курсу в обсязі –90 год.**

(3 кредити ECTS) зокрема: лекції – 8 год., консультації – 1 год., самостійна робота 81 год. У курсі передбачено 2 частини. Завершується дисципліна – заліком.

В результаті вивчення навчальної дисципліни студент повинен:

Знати основні моделі для аналізу алгоритмів та визначенню їх складності.

Вміти провести аналіз алгоритму та оцінити його складність.

4. Завдання (навчальні цілі):

Набуття знань, умінь та навичок (компетентностей) на рівні новітніх досягнень у програмуванні та обробці великих об’ємом даних, відповідно освітньої кваліфікації «Бакалавр з комп’ютерних наук». Зокрема, розвивати

- здатність до 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;
- здатність до інтелектуального багатовимірного аналізу даних та їхньої оперативної аналітичної обробки з візуалізацією результатів аналізу в процесі розв’язування прикладних задач у галузі комп’ютерних наук;
- здатність до алгоритмічного мислення.

У ході навчання студенти ознайомлюються з основними моделями для аналізу алгоритмів та визначенню їх складності.

5. Результати навчання за дисципліною:

Результат навчання (1. знати; 2. вміти; 3. комунікація; 4. автономність та відповідальність)		Форми (та/або методи і технології) викладання і навчання	Методи оцінювання та пороговий критерій оцінювання (за необхідності)	Відсоток у підсумковій оцінці з дисципліни
Код	Результат навчання			
PH1.1	Знати моделі для аналізу та оцінки складності алгоритмів	Лекція, самостійна робота	Контрольна робота 1	16%
PH1.2	Знати складності основних алгоритмів	Лекція, самостійна робота	Контрольна робота 1	15%
PH1.3	Знати основні підходи до проектування великих систем	Лекція, самостійна робота	Контрольна робота 2	20%
PH2.1	Вміти провести аналіз алгоритму та оцінити його складність	Лекція, самостійна робота	Контрольна робота 1, 2	23%
PH3.1	Обґрунтовувати власний погляд на задачу, спілкуватися з колегами з питань проектування та розробки програмних систем, складати письмові звіти	Лекція, самостійна робота	Контрольна робота 1, 2	10%
PH4.1	Організовувати свою самостійну роботу для досягнення результату	Самостійна робота	Контрольна робота 1,2	8%
PH4.2	Відповідально ставитися до виконуваних робіт, нести відповідальність за їх якість	Самостійна робота	Контрольна робота 1,2	8%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни	PH 1.1	PH 1.2	PH 1.3	PH 2.1	PH 3.1	PH 4.1	PH 4.2
Програмні результати навчання							
(з опису освітньої програми)							
ПРН18.1. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.	+	+	+	+	+	+	+

7. Схема формування оцінки.

7.1. Форми оцінювання студентів:

- семестрове оцінювання:

1. Контрольна робота 1: PH1.1, PH 1.2, PH2.1, PH3.1, PH4.1, PH4.2 - 50 балів/30 бали

2. Контрольна робота 2: PH1.3, PH2.1, PH3.1, PH4.1, PH4.2 - 50 балів/30 бали

- підсумкове оцінювання у формі заліку. Виставляється за результатами роботи студентами впродовж усього семестру та не передбачає додаткових заходів оцінювання для успішних студентів.

Запитання для підготовки до контрольних робіт

1. Параметри оцінки складності алгоритмів.
2. Особливості моделі ДАМ.
3. Модель ДАМП та порівняння з моделлю ДАМ.
4. Багатострічкова машина Т'юринга.
5. Методика оцінки складності програм на мовах високого рівня.
6. Оцінка складності рекурсивних програм.
7. Алгоритми на графах та оцінки їх складності.
8. Оцінка складності для множення матриць за алгоритмом Штрассена.
9. Обернення матриць.
10. НВП-розклад матриць.
11. Недетермінована машина Т'юринга.
12. NP-повні задачі.
13. Приклади NP-повних задач.

7.2 Організація оцінювання:

Терміни проведення форм оцінювання:

1. Контрольна робота 1: до останнього тижня семестру.
2. Контрольна робота 2: до останнього тижня семестру.

Студент має право на одне перескладання кожної контрольної роботи із можливістю отримання максимально 80% початково визначених за цю контрольну роботу балів. Термін перескладання визначається викладачем.

7.3 Шкала відповідності оцінок

Зараховано / Passed	60-100
Не зараховано / Fail	0-59

8. Структура навчальної дисципліни. Тематичний план лекцій і лабораторних занять

№ лекції	Назва лекції	Кількість годин		
		Лекції		Самост. робота
Частина 1				
Моделі для аналізу та оцінки складності алгоритмів				
1	Тема 1. Поняття про складність алгоритмів. Моделі ДАМ і ДАМП та зв'язок між ними.			6
2	Тема 2. Багатострічкова машина Т'юринга. Зв'язки з моделями ДАМ і ДАМП.			5
3	Тема 3. Мови високого рівня та оцінки складності програм.			5
4	Тема 4. Множини, їх представлення. Оцінки складності для операцій над множинами.			5
5	Тема 5. Рекурсія. Методика оцінки складності рекурсивних алгоритмів. Динамічне програмування.			5
6	Тема 6. Алгоритми сортування. Оцінки їх складності.			10
7	Тема 7. Алгоритми на графах. Остовне дерево найменшої вартості. Пошуки в глибину та в ширину.			10
Контрольна робота 1				10
Частина 2				
Алгоритми для дій над матрицями. NP-повні задачі.				
8	Тема 8. Множення та обернення матриць. Алгоритм Штрассена. НВП-розклад матриць.			5
9	Тема 9. Недетермінована машина Т'юринга. NP-повні задачі. Приклади.			10
Контрольна робота 2				10
ВСЬОГО		8		81

Загальний обсяг 90 години, в тому числі:

Лекцій – **8 годин**.

Консультації – **1 год**.

Самостійна робота - **81 год**.

9. Рекомендовані джерела:

Основні:

1. Анисимов А.В. Рекурсивные преобразователи информации. 1987.
2. Ахо А., Хопкрофт Дж., Ульман Дж. Построение и анализ вычислительных алгоритмов. Мир. 1979.
3. Рейнгольд Э., Нивергельт Ю., Део Н. Комбинаторные алгоритмы. Мир. 1980.

Додаткові:

4. Грэхем Р., Кнут Д., Паташник О. Конкретная математика. Основание информатики. Мир. 1998. – 703с.
5. Вергунова І.М. Побудова та аналіз алгоритмів. Лекції. – Вінниця:ТВОРИ, 2020. – 164 с.