

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ КОМП'ЮТЕРНИХ НАУК ТА КІБЕРНЕТИКИ  
КАФЕДРА ІНТЕЛЕКТУАЛЬНИХ ПРОГРАМНИХ СИСТЕМ**

**«ЗАТВЕРДЖУЮ»**

Заступник декана  
з навчальної роботи

\_\_\_\_\_ Кашпур О.Ф.

«\_\_\_» \_\_\_\_\_ 2019 року

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ  
ОБЧИСЛЮВАЛЬНА ГЕОМЕТРІЯ ТА  
КОМП'ЮТЕРНА ГРАФІКА  
для студентів**

галузь знань	<b>12 Інформаційні технології</b>
спеціальність	<b>121 Інженерія програмного забезпечення</b>
освітній рівень	<b>бакалавр</b>
освітня програма	<b>Програмна інженерія</b>
блок вибору	<b>Програмна інженерія</b>
вид дисципліни	<b>вбіркова</b>

Форма навчання	<b>денна</b>
Навчальний рік	<b>2021/2022</b>
Семестр	<b>6</b>
Кількість кредитів ECTS	<b>5</b>
Мова викладання, навчання та оцінювання	<b>українська</b>
Форма заключного контролю	<b>іспит</b>

Викладачі: **к.ф.-м.н., доц. Шкільняк О.С.** (лекції)  
**асистент Стовба В.О.** (лабораторні заняття)

Пролонговано: на 20\_\_/20\_\_ н.р. \_\_\_\_\_ (\_\_\_\_\_) «\_\_»\_\_ 20\_\_р.  
(підпис, ПІБ, дата)

на 20\_\_/20\_\_ н.р. \_\_\_\_\_ (\_\_\_\_\_) «\_\_»\_\_ 20\_\_р.  
(підпис, ПІБ, дата)

**КИЇВ 2019**

Розробник: Шкільняк Оксана Степанівна, к. ф.-м. н., доцент, доцент кафедри інтелектуальних програмних систем.

ЗАТВЕРДЖЕНО

Завідувач кафедри інтелектуальних програмних систем

\_\_\_\_\_ О.І. Провотар

Протокол № \_\_ від «\_\_» \_\_\_\_\_ 2019 р.

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

Протокол від «\_\_» \_\_\_\_\_ 2019 року №\_\_

Голова науково-методичної комісії \_\_\_\_\_ Л.Л. Омельчук

«\_\_\_\_\_» \_\_\_\_\_ 2019 року

Затверджено вченою радою факультету комп'ютерних наук та кібернетики

Протокол від «\_\_» \_\_\_\_\_ 2019 року №\_\_

Голова вченої ради факультету \_\_\_\_\_ А.В. Анісімов

**1. Мета дисципліни** – детальне вивчення алгоритмів обчислювальної геометрії як базового апарату комп'ютерної графіки, що включає в себе розгляд низки різноманітних підходів та алгоритмів для розв'язання задач опуклості, перетину, геометричного пошуку, близькості, оптимізації.

## **2. Попередні вимоги до опанування навчальної дисципліни**

1. **Знати:** основи загальної алгебри та аналітичної геометрії, основи математичного аналізу, основи теорії алгоритмів, основні поняття складності обчислень, концепцію об'єктно-орієнтованого програмування.

2. **Вміти:** програмувати на мові високого рівня.

## **3. Анотація навчальної дисципліни**

Навчальна дисципліна «Обчислювальна геометрія та комп'ютерна графіка» є складовою освітньо-професійної програми підготовки фахівців за першим (бакалаврським) рівнем вищої освіти галузі знань 12 Інформаційні технології за спеціальністю 121 Інженерія програмного забезпечення освітньо-професійної програми „Програмна інженерія”.

Дисципліна належить до вибіркових дисциплін блоку вибору “Програмна інженерія”. Викладається у **6 семестрі 3 курсу в обсязі – 150 год. (5 кредитів ECTS)**, зокрема: лекції – 28 год., лабораторні заняття – 28 год., консультації – 2 год., самостійна робота – 64 год. У курсі передбачено 3 частини та 3 контрольні роботи. Завершується дисципліна – **іспитом в 6 семестрі**.

В результаті вивчення навчальної дисципліни студент повинен:

**знати** основні поняття обчислювальної геометрії як базового апарату комп'ютерної графіки; постановку різних типів задач обчислювальної геометрії та основні підходи і методи для їх розв'язання.

**вміти** застосовувати оптимальний алгоритм при розв'язанні задач обчислювальної геометрії, правильно впізнавати задачі обчислювальної геометрії в постановках задач комп'ютерної графіки, за необхідності будувати власні алгоритми, що базуються на модифікації відомих технік, обґрунтовувати їх складність та реалізовувати; обґрунтовувати власний погляд на розв'язання задачі, спілкуватися з колегами з питань проектування та розробки програм, складати письмові звіти.

Дисципліна є логічним продовженням, доповненням та розширенням обов'язкової дисципліни «Алгоритми та складність» освітньо-професійної програми „Програмна інженерія”.

**4. Завдання (навчальні цілі).** Основними завданнями дисципліни «Обчислювальна геометрія та комп'ютерна графіка» є набуття знань, умінь та навичок (компетенцій) на рівні новітніх досягнень в обчислювальній геометрії та комп'ютерній графіці відповідно до освітньої кваліфікації бакалавр з програмної інженерії.

Зокрема, розвивати:

- Здатність до абстрактного мислення, аналізу та синтезу (ЗК01).
- Здатність застосовувати знання у практичних ситуаціях (ЗК02).
- Здатність спілкуватися державною мовою як усно, так і письмово (ЗК03).
- Здатність вчитися і оволодівати сучасними знаннями (ЗК05).
- Здатність до пошуку, оброблення та аналізу інформації з різних джерел (ЗК06).
- Здатність до алгоритмічного та логічного мислення (СК14).
- Здатність застосовувати дискретні структури і сучасні методи дискретної математики під час аналізу, синтезу та проектування інформаційних систем різної природи (СК15.2).

## 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			
РН1.1	<i>Знати</i> основні поняття обчислювальної геометрії та комп'ютерної графіки.	<i>Лекція, лабораторне заняття</i>	<i>Контрольна робота, іспит</i>	40%
РН1.2	<i>Знати</i> постановку різних типів задач обчислювальної геометрії та основні підходи і методи для їх розв'язання	<i>Лекція, лабораторне заняття, самостійна робота</i>	<i>Контрольна робота, іспит</i>	
РН2.1	<i>Вміти</i> застосовувати оптимальний алгоритм при розв'язанні задач обчислювальної геометрії, правильно впізнавати задачі обчислювальної геометрії в постановках задач комп'ютерної графіки, за необхідності будувати власні алгоритми, що базуються на модифікації відомих технік і обґрунтовувати їх складність.	<i>Лабораторне заняття, самостійна робота</i>	<i>Контрольна робота, поточне оцінювання, захист лабораторної роботи, іспит</i>	45%
РН2.2	<i>Вміти</i> реалізовувати розроблені алгоритми.	<i>Лабораторне заняття, самостійна робота</i>	<i>Захист лабораторної роботи</i>	5%
РН3.1	<i>Обґрунтовувати</i> власний погляд на розв'язання задачі, спілкуватися з колегами з питань проектування та розробки програм, складати письмові звіти	<i>Лабораторне заняття, самостійна робота</i>	<i>Поточне оцінювання, захист лабораторної роботи</i>	5%
РН4.1	<i>Відповідально</i> ставитися до виконуваних робіт, нести відповідальність за їх якість	<i>Лабораторне заняття</i>	<i>Захист лабораторної роботи</i>	5%

## 6. Співвідношення результатів навчання дисципліни із програмними результатами навчання.

Результати навчання дисципліни	Програмні результати навчання					
	РН1.1	РН1.2	РН2.1	РН2.2	РН3.1	РН4.1
<b>ПРН01.</b> Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки.		+	+		+	
<b>ПРН05.</b> Знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізу та математичного моделювання для розробки програмного забезпечення.	+	+	+	+		
<b>ПРН06.</b> Уміння вибирати та використовувати відповідну задачі методологію створення			+		+	+

програмного забезпечення.						
<b>ПРН13.</b> Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.		+	+	+		
<b>ПРН25.2.</b> Аналізувати, оцінювати і вибирати інструментальні та обчислювальні засоби, технології, алгоритмічні і програмні рішення для розв'язання завдань інженерії програмного забезпечення.			+		+	+
<b>ПРН26.2.</b> Знати та мати навички реалізації основних алгоритмів та структур даних програмування.		+	+	+		
<b>ПРН28.2.</b> Знати та вміти застосовувати методи та алгоритми обчислювальної геометрії й комп'ютерної графіки.	+	+	+	+		

## 7. Схема формування оцінки.

### 7.1. Форми оцінювання студентів.

#### – семестрове оцінювання (максимальна кількість балів):

1. Контрольна робота 1: РН 1.1, РН 1.2, РН 2.1 – **14 б./8 б.**
2. Контрольна робота 2: РН 1.1, РН 1.2, РН 2.1 – **18 б./11 б.**
3. Контрольна робота 3: РН 1.1, РН 1.2, РН 2.1 – **11 б./5 б.**
4. Лабораторний проект: РН 2.1-2.2, РН 3.1, РН 4.1 – **20 б./12 б.**

#### Підсумкове оцінювання (у формі іспиту):

- максимальна кількість балів які можуть бути отримані студентом: 40 балів;
- результати навчання які будуть оцінюватись: РН 1.1, РН 1.2, РН 2.1
- форма проведення і види завдань: письмово-усна форма, 5 завдань по 8 балів (20%) кожне.

Для отримання загальної позитивної оцінки з дисципліни оцінка за іспит не може бути меншою **24 балів**.

Здобувач освіти може бути **недопущений** до підсумкового оцінювання, якщо під час семестру він:

- 1) не досяг мінімального порогового рівня (60%) оцінки тих результатів навчання, які не можуть бути оцінені під час підсумкового контролю;
- 2) набрав кількість балів, що є недостатньою для отримання позитивної оцінки навіть у випадку досягнення ним на підсумковому контролі максимально можливого результату.

### 7.2. Організація оцінювання.

#### Терміни проведення форм оцінювання:

1. Контрольна робота 1: після 4-го лекційного заняття.
2. Контрольна робота 2: після 10-го лекційного заняття.
3. Контрольна робота 3: після 14-го лекційного заняття.
4. Лабораторний проект: контроль через одне лабораторне заняття.

Студент має право один раз перескласти контрольну роботу з можливістю отримати не більше 80% балів, призначених за роботу. Термін перескладання визначається викладачем.

У випадку відсутності студента з поважних причин відпрацювання та перездачі контрольних робіт здійснюються у відповідності до „Положення про порядок оцінювання знань студентів при кредитно-модульній системі організації навчального процесу” від 1 жовтня 2010 року.

У разі неякісного або невчасного виконання лабораторної роботи викладач має право не зарахувати завдання або знизити за нього бали.

### 7.3. Шкала відповідності оцінок

<b>Відмінно / Excellent</b>	90-100
<b>Добре / Good</b>	75-89
<b>Задовільно / Satisfactory</b>	60-74
<b>Незадовільно / Fail</b>	0-59

### 8. Структура навчальної дисципліни. Тематичний план лекцій.

№ лекції	Назва лекції	Кількість годин		
		Лекції	Лаб.	С/р
<b>Частина 1. Геометричний пошук</b>				
1	<b>Тема 1.</b> Вступ до предмету. Основні означення та структури даних.	2	2	5
2	<b>Тема 2.</b> Поняття геометричного пошуку. Основні моделі геометричного пошуку.	2	2	6
3	<b>Тема 3.</b> Задачі локалізації точки.	2	2	6
4	<b>Тема 4.</b> Задачі регіонального пошуку.	2	2	6
	Контрольна робота № 1			2
		8	8	25
<b>Частина 2. Побудова опуклої оболонки</b>				
5	<b>Тема 5.</b> Задача побудови опуклої оболонки. Метод Грехема. Метод Джарвіса.	2	2	6
6-7	<b>Тема 6.</b> Швидкі методи побудови опуклої оболонки.	4	4	13
8-9	<b>Тема 7.</b> Динамічні методи побудови опуклої оболонки.	4	4	12
10	<b>Тема 8.</b> Побудова опуклої оболонки в 3D.	2	2	6
	Контрольна робота № 2			2
		12	12	39
<b>Частина 3. Близькість та перетин</b>				
11	<b>Тема 9.</b> Постановка основних задач. Пошук найближчої пари методом «розділяй та владарюй».	2	2	6
12	<b>Тема 10.</b> Діаграма Вороного. Властивості та побудова.	2	2	7
13	<b>Тема 11.</b> Триангуляція Делоне.	2	2	7

14	<b>Тема 12.</b> Задачі перетину.	2	2	6
	Контрольна робота № 3			2
		8	8	28
	<b>ВСЬОГО</b>	<b>28</b>	<b>28</b>	<b>92</b>

Загальний обсяг– *150 год.*, у тому числі:

Лекцій – *28 год.*,

Консультацій – *2 год.*,

Лабораторних занять – *28 год.*,

Самостійна робота – *92 год.*

### **Теми, винесені на самостійне вивчення**

Регуляризація графа у методі ланцюгів.

Недоліки та вдосконалення методів Грехема та Джарвіса, метод Чана.

Версія методу побудови типу «розділяй та владарюй» з попереднім сортуванням точок.

Узагальнення алгоритмів побудови опуклої оболонки на вищі розмірності.

Оцінки складності задач на близькість.

Узагальнення діаграми Вороного.

Обґрунтування складності алгоритмів триангуляції Делоне. Метод замітаючої оболонки.

Знаходження перетину опуклих багатокутників. Пошук ядра багатокутника.

### **Перелік питань для підготовки до іспиту**

1. Роль обчислювальної геометрії для комп'ютерної графіки. Основні класи та типи задач. Застосування.
2. Напрямки зображувальної інформації та комп'ютерної графіки.
3. Розкрити поняття оцінки складності алгоритму.
4. Метод звідності задач. Визначення верхніх та нижніх оцінок (твердження).
5. Задачі прототипи. Нижні оцінки складності задач.
6. Загальні означення обчислювальної геометрії.
7. Структури даних. Множини, списки, черги.
8. Дерево відрізків. Операції вставки та вилучення інтервалів.
9. Реберний список з подвійними зв'язками.
10. Метод плоского замітання.
11. Поняття геометричного пошуку. Міри ефективності. Моделі геометричного пошуку. Типи пошукових запитів.
12. Метод векторного домінування. Приклади застосування
13. Застосування методу локусів до розв'язання задач регіонального пошуку.
14. Локалізація точки на планарному розбитті. Розв'язання задачі про приналежність простому багатокутнику.
15. Локалізація точки на планарному розбитті. Розв'язання задачі про приналежність опуклому багатокутнику.
16. Метод смуг. Оцінки складності.
17. Метод ланцюгів. Оцінки складності.
18. Регуляризація графа у методі ланцюгів. Оцінки складності.
19. Метод деталізації триангуляції. Оцінки складності.
20. Метод трапецій. Оцінки складності.

21. Регіональний пошук. Основні типи дій. Оцінки складності. Двовимірний випадок.
22. Регіональний пошук. Метод 2-d дерева.
23. Регіональний пошук. Метод дерева регіонів.
24. Техніка fractional cascading. Покращення методу дерева регіонів.
25. Регіональний пошук у просторах вищих розмірностей.
26. Опуклі оболонки. Основні поняття. Постановка та схема розв'язання основних задач. Нижня оцінка задачі побудови опуклої оболонки множини точок.
27. Метод Грехема. Оцінка складності.
28. Метод Джарвіса. Оцінка складності.
29. Метод Чана.
30. Швидкі методи побудови опуклої оболонки. Швидкобол. Оцінка складності.
31. Швидкі методи побудови опуклої оболонки. "Розділяй та владарюй". Оцінка складності.
32. Динамічні алгоритми побудови опуклої оболонки.
33. Відкритий алгоритм Препарати. Інкрементний алгоритм. Оцінки складності.
34. Алгоритм динамічної підтримки опуклої оболонки. Оцінка складності.
35. Алгоритм апроксимації опуклої оболонки. Оцінка складності.
36. Опукла оболонка простого многокутника. Оцінка складності.
37. Узагальнення алгоритмів побудови опуклої оболонки на вищі розмірності.
38. Метод «загортання подарунку» в 3D.
39. Алгоритм типу «розділяй та владарюй» в 3D.
40. Рандомізований інкрементний алгоритм в 3D.
41. Близькість. Постановка основних задач.
42. Близькість. Обґрунтування нижніх оцінок складності основних задач.
43. Найближча пара - метод «розділяй та владарюй».
44. Означення та властивості діаграми Вороного.
45. Методи побудови діаграми Вороного. Інкрементний алгоритм. Складність.
46. Методи побудови діаграми Вороного. Метод Форчуна. Складність.
47. Методи побудови діаграми Вороного. Метод «розділяй та владарюй». Складність.
48. Розв'язання задач близькості за допомогою діаграми Вороного.
49. Триангуляція Делоне, її властивості.
50. Алгоритми триангуляції Делоне: метод заміни ребра, інкрементний, підхід «розділяй та владарюй» та їх складність.
51. Зв'язок триангуляції Делоне, діаграми Вороного і опуклої оболонки множини точок.
52. Перетин. Постановка основних задач.
53. Пошук перетинів відрізків методом плоского замітання.
54. Перевірка перетину простих многокутників та простоти многокутника.
55. Перетин півплощин.
56. Знаходження перетину опуклих многокутників.
57. Пошук ядра многокутника.

## 9. Рекомендовані джерела

### Основні

1. Ф. Препарата, М. Шеймос. Вычислительная геометрия: введение. – М.: Мир, 1989.
2. А. Ахо, Дж. Хопкрофт, Дж. Ульман. Построение и анализ вычислительных алгоритмов. – М.: Мир, 1979.
3. В.М. Терещенко, І.В. Кравченко, А.В. Анісімов. Основні алгоритми обчислювальної геометрії. – К., 2002.

4. S.L. Devadoss, J. O'Rourke. Discrete and Computational Geometry. Princeton University Press, 2011.
5. M. de Berg, O. Cheong, M. van Kreveld, M. Overmars. Computational Geometry: Algorithms and Applications. 3rd edition. – Springer, 2008.

#### **Додаткові**

6. J. O'Rourke. Computational Geometry in C. Cambridge University Press, Second Edition, 1998.
7. David M. Mount. Lecture notes for the course CMSC 754 Computational Geometry (<https://www.cs.umd.edu/class/fall2014/cmsc754/Lects/cmsc754-fall14-lects.pdf> )
8. D.A. Sinclair. S-hull: a fast radial sweep-hull routine for Delaunay triangulation ([http://www.s-hull.org/paper/s\\_hull.pdf](http://www.s-hull.org/paper/s_hull.pdf) )
9. М. Ласло. Вычислительная геометрия и компьютерная графика на C++. – М.: Бином, 1997.