

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**МЕХАНІКО-МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ
Кафедра математичної фізики**

«ЗАТВЕРДЖУЮ»
Заступник декана
з навчальної роботи
Харитонов О.М.
«серпень» 2021 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

**Об'єктно-орієнтоване програмування
для студентів**

галузь знань	11 «Математика та статистика»
спеціальність	111 «Математика»
освітній рівень	перший (бакалавр)
освітня програма	«Математика»
вид дисципліни	обов'язкова

Форма навчання	денна
Навчальний рік	2021/2022
Семестр	2
Кількість кредитів ECTS	4
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	залік

Викладачі: Крєневич Андрій Павлович к.ф.-м. н., доцент кафедри математичної фізики;
Довгий Борис Павлович, к.ф.-м. н., доцент кафедри математичної фізики; Бородин Віктор
Анатолійович, к.ф.-м. н., доцент кафедри математичної фізики; Гап'як Ігор Васильович,
к.ф.-м. н., асистент кафедри математичної фізики; Клевцовський Арсен Володимирович,
к.ф.-м. н., асистент кафедри математичної фізики

Пролонговано: на 20 /20 н.р. () « » 20 р.
на 20 /20 н.р. () « » 20 р.

КИЇВ – 2021

Розробник¹: Кренивч Андрій Павлович, к.ф.-м. н., доцент кафедри математичної фізики.

ЗАТВЕРДЖЕНО

Зав. кафедри математичної фізики

 Самойленко В.Г.

Протокол №8 від 9 лютого 2021 р.

Схвалено науково-методичною комісією механіко-математичного факультету

Протокол від “30” березня 2021 року №8

Голова науково-методичної комісії  професор, д.ф.-м.н. Олійник А.С.

¹ Розробляється лектором. Робоча програма навчальної дисципліни розглядається на засіданні кафедри, науково-методичної комісії факультету/інституту, підписується завідувачем кафедри, головою науково-методичної комісії факультету/інституту і затверджується заступником декана/директора інституту з навчальної роботи.

1. Мета дисципліни

Мета дисципліни – ознайомлення та оволодіння сучасними методами та теоретичними положеннями, притаманними інформатиці та програмній інженерії, та їхнє застосування при побудові алгоритмів та програм для сучасних комп'ютерів.

2. Попередні вимоги до опанування або вибору навчальної дисципліни.

Для успішного опанування курсу, студент повинен знати основні поняття інформатики – виконавець, алгоритм, програма; синтаксис, семантику та властивості структур керування (ланцюг, розгалуження, цикли); синтаксичні конструкції мови програмування Python; формальні методи побудови алгоритмів та програм за допомогою рекурентних співвідношень; основні властивості програм; будову простих та складених типів даних; опис та використання підпрограм; поняття про виключні ситуації; роботу з файлами; модульне програмування.

3. Анотація навчальної дисципліни:

Навчальна дисципліна «Об'єктно-орієнтоване програмування» є складовою освітньої програми підготовки фахівців за освітнім рівнем «бакалавр» галузі знань 11 математика та статистика зі спеціальності 111 математика освітньої програми «математика».

Дана дисципліна є обов'язковою.

Викладається у 2 семестрі 1 курсу в обсязі 120 год. (4 кредити ECTS) зокрема: лекції – всього 16 год., лабораторні 30 год., самостійна робота 66 год. У курсі передбачено 1 змістовий модуль та 1 модульна контрольна робота. Завершується дисципліна заліком.

4. Завдання (навчальні цілі):

формування здатності розв'язувати складні задачі та практичні проблеми у математиці або у процесі навчання, що передбачає застосування теорій та методів математики, статистики й комп'ютерних технологій і характеризується комплексністю та невизначеністю умов; набуття знань, умінь та навичок (компетентностей) на рівні новітніх досягнень у математиці, відповідно до освітнього рівня «Бакалавр». Зокрема, професійне оволодіння компетентностями:

- 1) Здатність до абстрактного мислення, аналізу та синтезу;
- 2) Здатність застосовувати знання у практичних ситуаціях;
- 3) Знання й розуміння предметної області та професійної діяльності;
- 4) Навички використання інформаційних і комунікаційних технологій;
- 5) Здатність учитися і оволодівати сучасними знаннями;
- 6) Здатність до пошуку, обробки та аналізу інформації з різних джерел;
- 7) Здатність приймати обґрунтовані рішення;
- 8) Здатність працювати автономно;
- 9) Здатність використовувати стандартні прийоми та методи математичних досліджень, проявляти творчий підхід, ініціативу;
- 10) Здатність формулювати проблеми математично та в символній формі з метою спрощення їхнього аналізу й розв'язання;
- 11) Здатність здійснювати міркування та виокремлювати ланцюжки міркувань у математичних доведеннях на базі аксіоматичного підходу, а також розташовувати їх у логічну послідовність, у тому числі відрізняти основні ідеї від деталей і технічних викладок;
- 12) Здатність до кількісного мислення;
- 13) Здатність застосовувати чисельні методи для дослідження математичних моделей;
- 14) Здатність до аналізу математичних структур, у тому числі до оцінювання обґрунтованості й ефективності використовуваних математичних підходів;
- 15) Здатність застосовувати спеціалізовані мови програмування та пакети прикладних програм;

- 16) Здатність використовувати обчислювальні інструменти для чисельних і символьних розрахунків;
- 17) Здатність формулювати складні задачі оптимізації та прийняття рішень й інтерпретувати їхні розв'язки в оригінальному контексті цих задач;
- 18) Здатність отримувати якісну інформацію на основі кількісних даних;
- 19) Здатність розробляти експериментальні та спостережні дослідження й аналізувати дані, отримані на їх основі;
- 20) Здатність пояснювати математичними термінами результати, отримані під час розрахунків.

5. Результати навчання за дисципліною:

Результат навчання (1, знати; 2, вміти; 3, комунікація)		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				
PH 1.1	основні поняття парадигми об'єктно-орієнтованого програмування (клас та об'єкт, абстрагування, інкапсуляція, наслідування та поліморфізм)	Лекція, лабораторне заняття, самостійна робота	Контрольна робота 1 (60% правильних відповідей), виконання завдань, винесених на самостійну роботу, семестровий проект.	15%	
PH 1.2	основи мови візуального моделювання UML.	Лекція, лабораторне заняття, самостійна робота		5%	
PH 1.3	спеціальні методи та перевантаження операторів	Лекція, лабораторне заняття, самостійна робота		10%	
PH 1.4	ітератори та генератори	Лекція, лабораторне заняття, самостійна робота		5%	
PH 1.5	виключення та виключні ситуації	Лекція, лабораторне заняття, самостійна робота		5%	
PH 1.6	поняття про абстрактні класи та їхнє застосування.	Лекція, лабораторне заняття, самостійна робота		3%	
PH 1.7	принципи побудови програм з графічним інтерфейсом користувача.	Лекція, лабораторне заняття, самостійна робота		7%	
PH 2.1	створювати класи на основі процесу абстрагування.	Лабораторне заняття, самостійна робота		10%	
PH 2.2	розробляти базову архітектуру об'єктно-орієнтованих програмних систем з використанням уніфікованої мови візуального моделювання UML	Лабораторне заняття, самостійна робота		5%	
PH 2.3	перевантажувати для власних класів арифметичні оператори та спеціальні функції.	Лабораторне заняття, самостійна робота		5%	
PH 2.4	описувати шаблони програмування, такі як ітератори та генератори.	Лабораторне заняття, самостійна робота		5%	
PH 2.5	створювати користувацькі класи виключень та застосовувати їх на практиці.	Лабораторне заняття, самостійна робота		5%	
PH 2.6	створювати програми з графічним інтерфейсом та програми, що використовують анімацію.	Лабораторне заняття, самостійна робота		5%	
PH 3.1	здатність обґрунтовувати власний погляд на задачу та формулювати робочі гіпотези, спілкуватися з колегами з питань застосування математичних методів та теорій	Лабораторне заняття, самостійна робота		Виконання завдань, винесених на самостійну роботу	5%
PH 3.2	вироблення навиків командної роботи	Самостійна робота		Виконання завдань, винесених на самостійну роботу, семестровий проект.	5%

PH 3.3	здатність вчасно та правильно виконувати поставлені задачі та звітувати про їхнє виконання	Самостійна робота	Виконання завдань, винесених на самостійну роботу, семестровий проект.	5%
--------	--	-------------------	--	----

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Програмні результати навчання	Результати навчання дисципліни															
	PH 1.1	PH 1.2	PH 1.3	PH 1.4	PH 1.5	PH 1.6	PH 1.7	PH 2.1	PH 2.2	PH 2.3	PH 2.4	PH 2.5	PH 2.6	PH 3.1	PH 3.2	PH 3.3
	(з опису освітньої програми)															
PH-5 - Мати навички використання спеціалізованих програмних засобів комп'ютерної та прикладної математики і використовувати інтернет-ресурси	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+
PH-21 - Розв'язувати типові задачі математичного аналізу, алгебри, диференціальних та інтегральних рівнянь, оптимізації за допомогою чисельних методів.	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+	+

7. Схема формування оцінки.

7.1. Форми оцінювання студентів:

- оцінювання впродовж навчального періоду:

1. Виконання завдань, винесених на самостійну роботу: PH1.1, PH1.2, PH1.3, PH1.4, PH1.5, PH1.6, PH1.7, PH2.1, PH2.2, PH2.3, PH2.4, PH2.5, PH2.6, PH3.1, PH 3.2, PH 3.3 – 40 балів/23 балів;

2. Контрольна робота 1: PH1.1, PH1.2, PH1.3, PH1.4, PH1.5, PH1.6, PH1.7, PH2.1, PH2.2, PH2.3, PH2.4, PH2.5, PH2.6 – 20 балів/12 балів;

3. Семестровий проект: PH1.1, PH1.2, PH1.5, PH1.7, PH2.1, PH2.2, PH2.5, PH2.6, PH3.1, PH3.3 – 40 балів/23 балів;

Разом має бути 100/60

- підсумкове оцінювання: залік.

7.2. Організація оцінювання:

Оцінювання студентів здійснюється за результатами виконання студентами лабораторних робіт, роботи студентів під час лабораторних занять, модульної контрольної роботи та захисту семестрового проекту.

Під час лабораторного заняття, при вивченні нової теми, викладач може здійснювати усне опитування студентів або викликати студентів до дошки для розбору типових задач. В результаті такого опитування, студент може отримати до 2 балів за заняття.

Протягом семестру студенти мають виконати 10 лабораторних робіт, кожна з яких може бути оцінена від 1 до 4 балів.

Модульна контрольна робота проводиться з використанням онлайн платформи у вигляді тесту та оцінюється за шкалою від 1 до 20 балів.

Загальна оцінка за семестр формується, як сума усіх балів, отриманих студентом протягом семестру відповідно до зазначених вище критеріїв. При цьому, якщо його оцінка перевищує 60 балів, то вона встановлюється 60 балів.

Семестровий проект полягає у розв'язанні комплексної задачі, що охоплює усі теми дисципліни. Захист проекту здійснюється на останньому семестровому занятті. Під час захисту

студент має продемонструвати викладачу виконаний семестровий проект. Оцінка проекту здійснюється за шкалою від 1 до 40 балів. Допускається виконання семестрового проекту групою студентів (не більше трьох). У такому разі має бути чітко розписано, яку ділянку роботи який студент виконав. При цьому викладач має індивідуально оцінити результат роботи кожного студента за шкалою від 1 до 40 балів.

Умовою досягнення позитивної оцінки за дисципліну є отримання оцінки не меншої за 60 балів, що є сумою оцінки за роботу протягом семестру та оцінки отриманою у результаті захисту семестрового проекту.

Терміни проведення форм оцінювання:

	Період для здійснення відповідної форма оцінювання
Модульна контрольна робота	10 й тиждень навчального періоду.
Захист семестрового проекту	останній тиждень навчального періоду

7.3. Шкала відповідності оцінок

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

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

№ п/п	Назва теми	У тому числі		
		Лекції	Лабораторні заняття	Самостійна робота
1	Вступ до ООП. Об'єкти та класи	4	6	16
2	Наслідування та поліморфізм	2	6	12
3	Спеціальні методи	2	4	6
4	Ітератори та генератори	1	2	6
5	Створення виключень	1	2	4
6	Абстрактні класи	2	2	6
7	Графічний інтерфейс та створення анімаційних програм	4	6	16
	Залік		2	

Загальний обсяг: 120 год., у тому числі:

Лекцій – 16 год.

Лабораторних занять – 30 год.

Консультацій – 8 год.

Самостійної роботи – 66 год.

9. Рекомендовані джерела

Основні:

1. Крєневич А.П. Python у прикладах і задачах. Частина 2. Об'єктно-орієнтоване програмування. Навчальний посібник – К.: ВПЦ "Київський Університет", 2020. – 152 с.

2. Крєневич А.П. Методичні вказівки до лабораторних занять із дисципліни "Об'єктно-орієнтоване програмування" для студентів механіко-математичного факультету – К.: ВПЦ "Київський Університет", 2019. – ___ с.

3. The Python Tutorial [Електронний ресурс] – Режим доступу до ресурсу: <https://docs.python.org/3/tutorial/index.html>.

4. Орлов С. А. Технологии разработки программного обеспечения. Разработка сложных программных систем [Текст] : учеб. пособие для вузов по направлению "Информатика и вычисл. техника" / Сергей Александрович Орлов. – СПб.: Питер, 2002. – 463 с.

5. Прохоренок Н. А. Python 3 и PyQt. Разработка приложений. / Николай Анатольевич Прохоренок. – СПб: БХВ-Петербург, 2012. – 704 с.
6. Васильев А. Н. Python на примерах. Практический курс по программированию / А. Н. Васильев. – СПб.: Наука и техника, 2016. – 432 с. – (Просто о сложном).

Додаткові:

7. Кренивич А. П. Python у прикладах і задачах Частина 1. Структурне програмування [Електронний ресурс] / Андрій Павлович Кренивич. – 2017. – Режим доступу до ресурсу: <http://www.matfiz.univ.kiev.ua/books..>
8. Кренивич, А.П. С у задачах і прикладах : навчальний посібник із дисципліни "Інформатика та програмування" / А.П. Кренивич, О.В. Обвінцев. – К. : Видавничо-поліграфічний центр "Київський університет", 2011. – 208 с.
9. Python 3 для начинающих [Електронний ресурс] – Режим доступу до ресурсу: pythonworld.ru.
10. Збірник задач з дисципліни "Інформатика і програмування" / Вакал Є.С., Личман В.В., Обвінцев О.В., Бублик В.В., Довгий Б.П., Попов В.В. -2-ге видання, виправлене та доповнене –К.: ВПЦ "Київський університет", 2006.– 94 с.
11. Абрамов С.А., Гнездилова Г.Г., Капустина Е.Н., Селюн М.И. Задачи по программированию. –М.: Наука, 1988. – 224 с.
12. Златопольский Д.М. Сборник задач по программированию. – 2-е издание, переработанное и дополненное. – СПб.: БХВ-Петербург, 2007. –240 с.
13. Пильщиков В.Н. Сборник упражнений по языку Паскаль: Учебное пособие для вузов . –М.: Наука, 1989. –160 с.
14. Проскуряков И.В. Сборник задач по линейной алгебре. 11-е издание, стереотипное. – СПб.: Лань, 2008. –480 с.
15. Вирт Н. Систематическое программирование. Введение.–М.: Мир, 1977. –184 с.
16. Вирт Н. Алгоритмы + структуры данных = программы.–М.: Мир, 1985. –406 с.
17. Навчальні матеріали: Python [Електронний ресурс] – Режим доступу до ресурсу: <http://www.matfiz.univ.kiev.ua/pages/13>.
18. E-Olymp [Електронний ресурс] – Режим доступу до ресурсу: www.e-olymp.com.
19. Школа программиста [Електронний ресурс] – Режим доступу до ресурсу: <http://acmp.ru/>