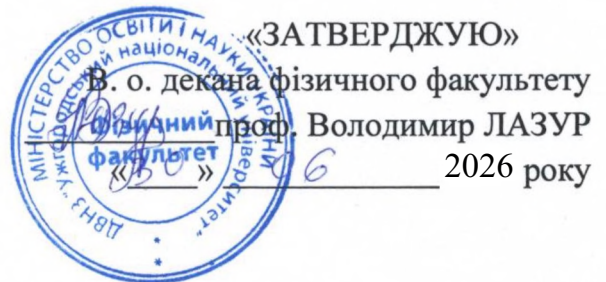


**ДЕРЖАВНИЙ ВИЩИЙ НАВЧАЛЬНИЙ ЗАКЛАД
“УЖГОРОДСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ”
ФІЗИЧНИЙ ФАКУЛЬТЕТ
Кафедра теоретичної фізики**



РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

**ІНЖЕНЕРІЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ ТА
ОБ’ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ**

Рівень вищої освіти	перший (бакалаврський)
Галузь знань	А Освіта
Спеціальність	А4 Середня освіта (за предметними спеціальностями)
Предметна спеціальність	А4.08 Середня освіта (Фізика та астрономія)
Освітня програма	«Фізика. Інформатика»
Статус дисципліни	обов’язкова
Мова навчання	українська

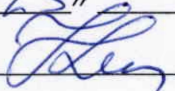
Ужгород – 2026

Робоча програма навчальної дисципліни **«Інженерія програмного забезпечення та об'єктно-орієнтоване програмування»** для здобувачів першого (бакалаврського) рівня вищої освіти галузі знань А Освіта, спеціальності А4 Середня освіта (за предметними спеціальностями), предметної спеціальності 014.08 Середня освіта (Фізика та астрономія), освітньої програми «Фізика. Інформатика».

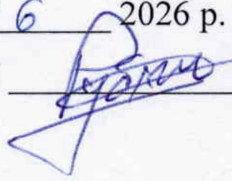
Розробники: Мисло Ю.М. – доцент кафедри твердотільної електроніки та інформаційної безпеки, кандидат фізико-математичних наук.
Мулеса П.П. – доктор педагогічних наук, доцент кафедри кібернетики і прикладної математики
Нодь Є. А. – к.ф.-м.н., доцент кафедри теоретичної фізики

Робочу програму розглянуто та затверджено на засіданні кафедри теоретичної фізики

протокол № 12 від «23» 06 2026 р.

Завідувач кафедри  Мирослав КАРБОВАНЕЦЬ

Схвалено науково-методичною комісією фізичного факультету
протокол № 8 від «30» 06 2026 р.

Голова науково-методичної комісії  Василь РУБИШ

1.ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Найменування показників	Розподіл годин за навчальним планом
	Денна форма навчання
Кількість кредитів ЄКТС – 6,5: 2-й семестр – 3,5; 3-й семестр – 3,0	Рік підготовки
Загальна кількість годин – 195 2-й семестр – 105 3-й семестр – 90	1-2-й
Кількість модулів – 4	Семестр
Тижневих годин для денної форми навчання: 2-й семестр аудиторних – 3 самостійної роботи студента – 3 3-й семестр аудиторних – 2 самостійної роботи студента – 2	2-й, 3-й
	Лекції
	38 год. 2-й семестр – 20 год. 3-й семестр – 18 год.
	Практичні (семінарські)
	–
	Лабораторні
	58 год. 2-й семестр – 32 год. 3-й семестр – 26 год.
Вид підсумкового контролю: 2-й семестр – залік; 3-й семестр – іспит.	Самостійна робота
Форма підсумкового контролю: усна	99 год. 2-й семестр – 53 год. 3-й семестр – 46 год.

2. МЕТА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Метою вивчення навчальної дисципліни «**Інженерія програмного забезпечення та об'єктно-орієнтоване програмування**» є формування у студентів знань про сучасні інформаційні технології та принципи роботи інформаційних і обчислювальних систем. Студенти повинні засвоїти поняття інформації, інформаційних процесів та способи її кодування в пам'яті ПК.

Важливим завданням є набуття практичних навичок роботи з програмним забезпеченням ПК, включаючи текстові процесори, електронні таблиці, бази даних, графічні редактори та засоби створення презентацій. Дисципліна передбачає ознайомлення з основами алгоритмізації та технологією програмування для вирішення прикладних задач. Окрему увагу приділено використанню хмарних технологій та засобів дистанційного навчання для ефективної обробки й передачі інформації.

Відповідно до освітньої програми, вивчення дисципліни сприяє формуванню у здобувачів вищої освіти таких компетентностей:

Інтегральна компетентності (ІК)

ІК. Здатність розв'язувати спеціалізовані практичні завдання в освітній галузі, що передбачає застосування концептуальних методів освітніх наук, предметних знань, інтеграції професійних та науково-дослідницьких знань з фізики та астрономії і інформатики, методики їх навчання і характеризується комплексністю та невизначеністю умов організації освітнього процесу в закладах загальної середньої освіти.

Загальні компетентності (ЗК):

ЗК1. Здатність до абстрактного мислення, аналізу та синтезу, до застосування знань у практичних ситуаціях.

ЗК4. Здатність орієнтуватися в інформаційному просторі, здійснювати пошук і критично оцінювати інформацію, оперувати нею в професійній діяльності.

ЗК12. Здатність вчитися і оволодівати сучасними знаннями.

Фахові компетентності спеціальності (ФК):

ФК1. Здатність перенесення системи наукових знань у професійну діяльність та в площину навчального предмету.

ФК4. Здатність формувати і розвивати в здобувачів освіти ключові компетентності і наскрізні вміння, визначені державними стандартами освіти; здійснювати інтегроване навчання здобувачів освіти; добирати і використовувати сучасні й ефективні методики і технології навчання, виховання й розвитку здобувачів освіти; формувати ціннісні ставлення в здобувачів освіти, розвивати критичне мислення.

Фахові (предметні) компетентності:

ПК6. Здатність використовувати знання наукових фактів, концепцій, теорій, принципів і методів сучасної інформатики у практиці навчання інформатики.

ПК7. Володіння методами інформаційного моделювання; здатність реалізовувати інформаційну модель засобами інформаційно-комунікаційних технологій; проводити комп'ютерний експеримент, інтерпретувати, аналізувати та узагальнювати його результати.

ПК8. Здатність ефективно використовувати наявні та створювати нові електронні (цифрові) ресурси сучасними мовами програмування.

ПК9. Здатність використовувати програмні засоби загального та спеціального призначення для розв'язання прикладних задач з інформатики.

ПК10. Володіння технологіями експлуатації комп'ютерної мережі; здатність реалізовувати комплекс заходів, спрямованих на забезпечення захищеності інформації, здатність формувати вміння безпечної роботи здобувачів освіти у комп'ютерній мережі.

ПК11. Здатність розв'язувати задачі шкільного курсу інформатики різного рівня складності, аналізувати та оцінювати ефективність розв'язку та формувати відповідні вміння в учнів.

ПК12. Здатність добирати та використовувати сучасні інформаційно-комунікаційні технології в освітньому процесі та в позакласній роботі, аналізувати й оцінювати доцільність й ефективність їх застосування.

ПК14. Здатність до створення концептуальної, логічної та фізичної моделей проектування систем керування базами даних.

3. ПЕРЕДУМОВИ ДЛЯ ВИВЧЕННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Передумовами вивчення навчальної дисципліни **«Інженерія програмного забезпечення та об'єктно-орієнтоване програмування»** є знання шкільного предмету інформатики та опанування таких навчальних дисциплін освітньої програми:

ОК 8. Комп'ютерна графіка;

ОК 9. Програмування і алгоритмізація;

ОК 36. Навчальна обчислювальна практика.

4. ОЧІКУВАНІ РЕЗУЛЬТАТИ НАВЧАННЯ

Відповідно до освітньої програми **“Фізика. Інформатика”**, вивчення навчальної дисципліни **“Інженерія програмного забезпечення та об'єктно-орієнтоване програмування”** повинно забезпечити досягнення здобувачами вищої освіти таких програмних результатів навчання:

Програмні результати навчання	Шифр ПРН
Здійснює добір і застосовує сучасні освітні технології та методики для формування предметних компетентностей учнів; критично оцінює результати їх навчання та ефективність уроку.	РН4
Демонструє знання основ фундаментальних і прикладних наук (відповідно до предметної спеціальності), оперує базовими категоріями та поняттями предметної області спеціальності.	РН7
Застосовує сучасні інформаційно-комунікаційні та цифрові технології у професійній діяльності.	РН9
Демонструє володіння сучасними технологіями пошуку наукової інформації для самоосвіти та застосування її у професійній діяльності.	РН10
Визначає структуру предметної галузі інформатики, її місце в системі наук, пояснює перспективи розвитку інформатики та інформаційних технологій, їхнє суспільне значення.	ПРН7

Знає та розуміє фізичні, логічні та математичні основи інформаційних технологій; пояснює та застосовує способи двійкового кодування текстової, числової, графічної, звукової та відеоінформації.	ПРН8
Використовує інформаційно-комунікаційні технології для подання, редагування, збереження та перетворення текстової, числової, графічної, звукової та відеоінформації.	ПРН9
Знає та розуміє принципи функціонування та основи архітектури комп'ютерних систем та мереж; обґрунтовує необхідність та використовує апаратне та програмне забезпечення для налагодження та адміністрування локальної мережі	ПРН10
Визначає та застосовує методи розроблення та дослідження алгоритмів розв'язування задач з інформатики, описує і застосовує методи оцінювання ефективності алгоритмів.	ПРН11
Аналізує та здатний розкривати дидактичний потенціал електронних засобів навчання, бере участь в організації дистанційного навчання з використанням систем його підтримки та електронних (цифрових) освітніх ресурсів.	ПРН13
Уміє реалізовувати алгоритми розв'язання задач мовами програмування, вибирати й застосовувати інформаційно-комунікаційні технології; розв'язує задачі шкільного курсу з інформатики різного рівня складності.	ПРН15
Розуміє і реалізує сучасні методики й освітні технології навчання інформатики для виконання освітньої програми в закладах загальної середньої освіти, застосовує інформаційно-комунікаційні технології на уроках і в позакласній роботі.	ПРН16

Очікувані результати навчання, які повинні бути досягнуті здобувачами освіти після опанування навчальної дисципліни **“Інженерія програмного забезпечення та об’єктно-орієнтоване програмування”**:

Очікувані результати навчання з дисципліни	Шифр ПРН
Здобувач вміє здійснювати добір і застосовувати сучасні освітні технології та методики для формування предметних компетентностей учнів; критично оцінювати результати їх навчання та ефективність уроку.	РН4
Здобувач демонструє знання основ фундаментальних і прикладних наук з інформатики, оперує базовими категоріями та поняттями предметної області спеціальності.	РН7
Здатний застосовувати сучасні інформаційно-комунікаційні та цифрові технології у професійній діяльності.	РН9
Здатний ефективно використовувати сучасні інформаційні технології для пошуку, аналізу та опрацювання наукових даних з метою самоосвіти та їх практичного застосування у професійній діяльності.	РН10
Здобувач здатний визначати структуру предметної галузі інформатики, її місце в системі наук, пояснювати перспективи розвитку інформатики та інформаційних технологій, їхнє суспільне значення.	ПРН7
Володіє знаннями про фізичні, логічні та математичні основи інформаційних технологій, уміє пояснювати принципи двійкового кодування та застосовувати їх для представлення текстової, числової, графічної, аудіо- та відеоінформації.	ПРН8

Здобувач здатний застосовувати інформаційно-комунікаційні технології для представлення, редагування, збереження та трансформації текстових, числових, графічних, аудіо- та відеоданих.	ПРН9
Здобувач знає та розуміє принципи функціонування та основи архітектури комп'ютерних систем та мереж; обґрунтовує необхідність та використовує апаратне та програмне забезпечення для налагодження та адміністрування локальної мережі	ПРН10
Здобувач здатний визначати та застосовувати методи розроблення та дослідження алгоритмів розв'язування задач з інформатики, описувати і застосовувати методи оцінювання ефективності алгоритмів.	ПРН11
Вміє використовувати електронні засоби та цифрові ресурси для організації дистанційного навчання.	ПРН13
Здатний реалізовувати алгоритми розв'язання задач мовами програмування, обирати та застосовувати інформаційно-комунікаційні технології для розв'язання задач з інформатики.	ПРН15
Володіє сучасними методиками та технологіями навчання інформатики, уміє застосовувати ІКТ у навчальному процесі та позакласній діяльності.	ПРН16

5. ЗАСОБИ ДІАГНОСТИКИ ТА КРИТЕРІЇ ОЦІНЮВАННЯ РЕЗУЛЬТАТІВ НАВЧАННЯ

Засоби оцінювання та методи демонстрування результатів навчання

Засобами оцінювання та методами демонстрування результатів навчання з навчальної дисципліни є:

- поточний контроль успішності,
- модульний контроль,
- захист лабораторних робіт,
- підсумковий контроль.

Форми поточного контролю та критерії оцінювання результатів навчання:

- вибіркове усне опитування;
- фронтальне стандартизоване усне та/або письмове опитування за основними питаннями теми заняття;
- тестування;
- перевірка якості виконання завдань для самостійної роботи, зокрема за конспектами матеріалів;
- оцінювання якості та повноти виконання завдань модульної контрольної роботи;
- оцінювання якості та повноти виконання лабораторних робіт.

Форма модульного контролю: поточне оцінювання та виконання модульної контрольної роботи у письмовій формі, сумарний результати яких оцінюються за 100-бальною шкалою за кожний модуль.

Форма підсумкового семестрового контролю: залік, екзамен. До заліку та екзамену допускаються студенти, які відпрацювали пропущені заняття і виконали модульні контрольні роботи.

Семестр 2

Розподіл балів, які отримують здобувачі вищої освіти (модуль 1)

Поточне оцінювання та самостійна робота				Модульна контрольна робота	Сума
T1	T2	T3	T4	50	100
12	12	13	13		

Розподіл балів, які отримують здобувачі вищої освіти (модуль 2)

Поточне оцінювання та самостійна робота				Модульна контрольна робота	Сума
T5	T6	T7	T8	50	100
12	12	13	13		

Оцінювання окремих видів навчальної роботи з дисципліни

Вид діяльності здобувача вищої освіти	Модуль 1		Модуль 2	
	Кількість	Максимальна кількість балів (сумарна)	Кількість	Максимальна кількість балів (сумарна)
Лабораторні заняття (допуск, виконання та захист)	4	50	4	50
Модульна контрольна робота	1	50	1	50
Разом	5	100	5	100

Семестр 3

Розподіл балів, які отримують здобувачі вищої освіти (модуль 1)

Поточне оцінювання та самостійна робота				Модульна контрольна робота	Сума
T9	T10	T11	T12	50	100
12	12	13	13		

Розподіл балів, які отримують здобувачі вищої освіти (модуль 2)

Поточне оцінювання та самостійна робота				Модульна контрольна робота	Сума
T13	T14	T15	T16	50	100
12	12	13	13		

Оцінювання окремих видів навчальної роботи з дисципліни

Вид діяльності здобувача вищої освіти	Модуль 1		Модуль 2	
	Кількість	Максимальна кількість балів (сумарна)	Кількість	Максимальна кількість балів (сумарна)
Лабораторні заняття (допуск, виконання та захист)	4	50	4	50
Модульна контрольна робота	1	50	1	50
Разом	5	100	5	100

Критерії оцінювання модульної контрольної роботи

При оцінюванні знань враховується в першу чергу повнота, правильність і вичерпність відповідей на поставлені в модульних контрольних роботах запитаннях. Максимальна кількість балів, що виставляється здобувачу вищої освіти за виконання контрольної роботи складає 50 балів.

41 – 50 балів виставляється, якщо під час проведення контролю було виявлено:

1. наявність у студента всебічних, повних, глибоких інтегрованих знань програмового матеріалу, вміння вільно виконувати завдання запропонованого варіанту;
2. вміння студента в письмовій та усній формі чітко, вичерпно і правильно викласти відповіді на питання запропонованого варіанту;
3. глибоке розуміння студентом взаємозв'язку головних понять і положень предмета, розуміння значення цих положень і понять для майбутньої професії;
4. високий рівень підготовленості студента з питань курсу до подальшої роботи над вдосконаленням рівня своєї професійної кваліфікації.

У відповідях студентів не має бути значних помилок. Робота виконана на 50 балів демонструє наявність у студента творчих здібностей.

26 – 40 балів виставляється, коли студент письмово відповів на всі запитання, засвоїв всю навчальну програму відповідного модуля. У відповідях можлива не більш як одна незначна помилка або виявлено декілька неточностей.

16 – 25 балів виставляється, коли студент дав відповіді на питання всіх завдань, але при цьому можуть проявитися певні прогалини у засвоєнні програми модуля. У відповідях можуть зустрітися не більше як одна груба помилка або декілька значних та істотних неточностей.

0 – 15 балів виставляється за роботу, яка засвідчує про наявність у студента великих та суттєвих прогалин у знаннях основного матеріалу модуля, а у наявних його письмових

відповідях є як принципові, так і грубі помилки. Студенти, які не представили письмові відповіді на модульних контрольних роботах, або не з'явилися на модульну контрольну роботу вважаються такими, що одержали 0 балів незалежно від причини невиконання (неявки).

Сумарна оцінка (від 0 до 100 балів) за модуль виставляється у відомість модульного контролю. Модуль зараховується, якщо сумарний бал складає не менше 60 балів, і виконані та зараховані всі завдання, які є складовими модуля.

Здобувач, який не з'явився на модульну контрольну роботу, або ж його модульна оцінка складає від 0 до 34 балів, повинен до проведення підсумкового семестрового контролю покращити цю оцінку принаймні до показника не менше 35 балів у строки, визначені викладачем дисципліни. Без такого покращення він до семестрового контролю не допускається.

Підсумкова модульна оцінка з даної навчальної дисципліни визначається як середнє арифметичне результатів двох модульних контролів та виставляється у відомість модульного контролю за 100-бальною шкалою, шкалою ЄКТС та національною шкалою (див. табл. «Шкала оцінювання: національна та ECTS»).

Критерії оцінювання підсумкового семестрового контролю

Підсумковий семестровий контроль з дисципліни **“Інженерія програмного забезпечення та об'єктно-орієнтоване програмування”** у першому семестрі здійснюється у формі заліку. Залік проводиться в усній формі шляхом співбесіди. Результати заліку оцінюються за двобальною шкалою: “зараховано” або “незараховано”. Підсумкова оцінка визначається наступними критеріями:

Оцінка “зараховано” - якщо студент достатньо чітко і грамотно відповідає на питання в межах матеріалу, викладеного у рамках лекційних занять, може показати та обґрунтувати взаємозв'язок різних частин матеріалу, пройденого у межах матеріалу навчальної дисципліни; демонструє здатність до мислення, при відповіді на питання розмірковує, спираючись на отримані у рамках курсу знання, не допускає істотних неточностей у відповіді, правильно вибудовує логіку вирішення типових завдань;

Оцінка “незараховано” - якщо студент викладає основні питання недостатньо чітко або допускає істотні помилки при їх викладі, не може пояснити зв'язків у рамках викладеного матеріалу, не знає значної частини програмного матеріалу, не може дати точних визначень понять, пройдених у рамках курсу, дає розпливчасті формулювання і не володіє в належній мірі термінологією, плутається при відповіді на додаткові питання, не володіє прийомами вирішення типових завдань.

Підсумковий семестровий контроль з дисципліни у другому семестрі здійснюється у формі іспиту. Іспит проводиться в усній формі шляхом співбесіди. Результати іспиту оцінюються за 5-ти бальною шкалою: “відмінно”, “добре”, “задовільно”, “незадовільно”.

Оцінка “відмінно” (А; 90-100) виставляється в тому разі, коли студент бездоганно оволодів всіма розділами програми, дав глибокі, чіткі і вичерпні відповіді на всі основні і додаткові запитання, виявив розуміння суті програмового матеріалу, вільно володів фактичним матеріалом та відповідним математичним апаратом, кваліфіковано використовував набуті знання для розв'язання конкретних практичних задач.

Оцінка “добре” (В, С; 74-89) виставляється тоді, коли студент виявив повне знання і розуміння програмового матеріалу, може використовувати набуті знання в практичній діяльності, дав вичерпні відповіді на всі запитання, але під час відповіді допускав окремі нечіткі формулювання і незначні неточності.

Оцінка “задовільно” (D, E; 60-73) виставляється в тому разі, коли студент в основному знає і розуміє фактичний матеріал курсу, дав в основному правильні відповіді на запитання, виявив уміння розібратися в усьому матеріалі курсу, але не може ґрунтовно пояснити окремі положення пройденого курсу, допускає неточності, недостатньо вміє застосовувати набуті знання для розв'язання конкретних практичних задач.

Оцінка “незадовільно” (FX, F; 0-59) виставляється тоді, коли студент не оволодів матеріалом даного курсу, виявив суттєві прогалини в знаннях основного програмового матеріалу, коли він під час відповіді на запитання виявив нерозуміння сутності основних понять та термінів навчальної дисципліни, допускає плутанину, не може застосовувати набуті знання для розв’язування конкретних практичних задач, тобто виявив відсутність мінімально необхідної кількості знань з даного курсу.

За бажанням студента, результуюча підсумкова оцінка може бути визначена, як інтегрована оцінка засвоєння всіх тем дисципліни і кількісно дорівнює середньому арифметичному балів, отриманих за кожний модуль. Переведення результатів, отриманих за 100-бальною шкалою оцінювання в національну 5-ти бальну та шкалу за системою ECTS здійснюється за наступною схемою:

Шкала оцінювання: національна та ECTS

Шкала ЄКТС	Диференційована шкала	Недиференційована шкала	Мін.бал- макс.бал
A	Відмінно	Зараховано	90-100
B	Добре		82-89
C			74-81
D	Задовільно		64-73
E			60-63
Fx F	Незадовільно	Не зараховано	35-59 0-34

Результати підсумкового контролю знань вносяться до відомості обліку успішності.

6. ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

6.1. Зміст навчальної дисципліни

СЕМЕСТР 2

Модуль 1 Інструментарій інженерії програмного забезпечення (ПЗ) та базове об'єктно-орієнтоване програмування (ООП)

Тема 1. Життєвий цикл розробки ПЗ (SDLC) та методології розробки.

Етапи створення програмного продукту (від аналізу вимог до підтримки). Каскадна (Waterfall) та гнучкі (Agile/Scrum) методології. Особливості ведення проєктів у науковій та освітній сферах.

Тема 2. Системи контролю версій (Git) та колективна розробка.

Робота з Git та GitHub. Створення гілок (branching), злиття (merging), вирішення конфліктів коду. Спільна робота над навчальними інженерними проєктами.

Тема 3. Концепція ООП. Класи та об'єкти в Python.

Перехід від процедурного підходу до об'єктного. Поняття класу як типу даних та об'єкта як екземпляру. Конструктор `__init__`, роль покажчика `self`. Створення класу `PhysicalBody` (Матеріальна точка) з атрибутами маси та координат.

Тема 4. Інкапсуляція та керування доступом до даних.

Захист даних від некоректного втручання. Публічні, захищені (`_`) та приватні (`__`) атрибути. Використання декоратора `@property` (геттери та сеттери). Автоматична валідація даних (наприклад, заборона встановлення від'ємної маси або швидкості, що перевищує швидкість світла).

Модуль 2. Поглиблене ООП та надійність програм

Тема 5. Успадкування та розширення моделей.

Ієрархія класів. Створення дочірніх класів на основі базових. Функція `super()`. Перевизначення методів. *Приклад*: Базовий клас `SpaceBody` $\rightarrow$ дочірні класи `Planet`, `Star`, `Asteroid`.

Тема 6. Поліморфізм та гнучкість інтерфейсів.

Концепція поліморфізму. Качина типізація (Duck Typing) в Python. Створення єдиного інтерфейсу для взаємодії з різними об'єктами. Обробка списку різних фізичних тіл (електронів, протонів) в одному циклі симуляції за допомогою спільного методу `calculate_force()`.

Тема 7. Магічні (dunder) методи та перевантаження операторів.

Керування поведінкою об'єктів за допомогою вбудованих методів (`__str__`, `__repr__`). Перевантаження математичних операторів (`__add__`, `__sub__`, `__mul__`). Створення повноцінного класу `Vector3D` для виконання векторних операцій у фізичних задачах без використання сторонніх бібліотек.

Тема 8. Сучасні стандарти якості коду та модульне тестування (Unit Testing).

Стандарт оформлення коду PEP 8. Статичний аналіз (лінери). Основи тестування за допомогою бібліотеки unittest або pytest. Написання тестів для перевірки правильності обчислення фізичних формул.

СЕМЕСТР 3

Модуль 1. Проєктування та архітектура програмного забезпечення

Тема 9. Принципи проєктування SOLID.

П'ять фундаментальних принципів об'єктно-орієнтованого проєктування (SOLID) простими словами. Як писати код, який легко розширювати (наприклад, додавати нові типи фізичних сил у симуляцію, не переписуючи старий код).

Тема 10. Шаблони проєктування (Design Patterns) у фізичному моделюванні.

Поняття про патерни. Породжуючі (Фабрика, Одинак), структурні та поведінкові патерни. Застосування патерну «Фабрика» для генерації великої кількості випадкових астрономічних об'єктів (зірок) у галактиці.

Тема 11. Абстрактні класи та інтерфейси.

Модуль abc (Abstract Base Classes). Проєктування архітектури великих програм за допомогою абстрактних методів. Створення шаблону для «Осцилятора» або «Оптичного приладу», який зобов'язує дочірні класи реалізувати певні фізичні закони.

Тема 12. Розробка графічного інтерфейсу користувача (GUI).

Створення візуальних віконних програм за допомогою бібліотеки CustomTkinter або PyQt. Робота з кнопками, текстовими полями, слайдерами. Навіщо вчителю GUI? (Для створення інтерактивних віртуальних лабораторних робіт, де учні можуть повзунками змінювати, наприклад, довжину нитки маятника чи напругу в колі).

Модуль 2. Прикладне інженерне застосування (GUI та Екосистема ПЗ)

Тема 13. Інтеграція обчислень та графіки в GUI-додатки.

Поєднання знань з першого курсу (Matplotlib/NumPy) з графічним інтерфейсом. Вбудовування динамічних графіків у вікно програми. Візуалізація результатів фізичного експерименту в реальному часі за натисканням кнопки у додатку.

Тема 14. Серіалізація даних та робота зі складними форматами (JSON/Бази даних).

Збереження стану об'єктів (серіалізація) за допомогою модуля json або pickle. Запис параметрів всієї фізичної системи у файл для подальшого завантаження. Коротке ознайомлення з легковагими БД (SQLite) для збереження результатів тестування учнів.

Тема 15. Документування проєкту та створення документації (Sphinx/Markdown).

Написання професійних інструкцій користувача та технічної документації. Автоматична генерація документації з коду за допомогою docstrings. Створення файлу README.md для власного програмного продукту.

Тема 16. Деплоймент, пакування та дистрибуція ПЗ.

Як передати свою програму іншим (наприклад, учням у школі, у яких не встановлений Python). Пакування коду в один виконуваний файл (.exe для Windows або .app для macOS) за допомогою утиліти **PyInstaller**.

6.2. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин					
	Форма навчання					
	у тому числі					
	Усього	лекції	практичні (семінарські)	лабораторні	індивідуальна робота	самостійна робота
2-й семестр						
Модуль 1						
Тема 1. Життєвий цикл розробки ПЗ (SDLC) та методології розробки.	11	2		3		6
Тема 2. Системи контролю версій (Git) та колективна розробка.	12	2		4		6
Тема 3. Концепція ООП. Класи та об'єкти в Python.	14	3		4		7
Тема 4. Інкапсуляція та керування доступом до даних.	14	3		4		7
Модульна контрольна робота	1			1		
Разом за модуль	52	10		16		26
Модуль 2						
Тема 5. Успадкування та розширення моделей.	12	2		4		6
Тема 6. Поліморфізм та гнучкість інтерфейсів.	13	2		4		7
Тема 7. Магічні (dunder) методи та перевантаження операторів.	14	3		4		7
Тема 8. Сучасні стандарти якості коду та модульне тестування (Unit Testing).	13	3		3		7
Модульна контрольна робота	1			1		
Разом за модуль	53	10		16		27
Разом за семестр	105	20		32		53
3-й семестр						
Модуль 1						
Тема 9. Принципи проєктування SOLID.	10	2		2		6
Тема 10. Шаблони проєктування (Design Patterns) у фізичному моделюванні.	11	2		3		6
Тема 11. Абстрактні класи та інтерфейси.	11	2		3		6
Тема 12. Розробка графічного інтерфейсу користувача (GUI).	11	2		3		6
Модульна контрольна робота	1			1		
Разом за модуль	44	8		12		24
Модуль 2						
Тема 13. Інтеграція обчислень та графіки в GUI-додатки.	10	2		3		5
Тема 14. Серіалізація даних та робота зі	10	2		3		5

складними форматами (JSON/Бази даних).						
Тема 15. Документування проєкту та створення документації (Sphinx/Markdown).	12	3		3		6
Тема 16. Деплоймент, пакування та дистрибуція ПЗ.	13	3		4		6
Модульна контрольна робота	1			1		
Разом за модуль	46	10		14		22
Разом за семестр	90	18		26		46
Усього	195	38		58		99

6.3. Теми лабораторних робіт

№ з/п	Назва теми	Кількість годин
Семестр 2. Від інженерних інструментів до фундаментального ООП		
1	Налаштування інженерного оточення та робота з Git	3
2	Проектування класу «Матеріальна точка» (Основи класів)	4
3	Захист даних та валідація у фізичних моделях (Інкапсуляція)	4
4	Ієрархія космічних об'єктів (Успадкування)	4
5	Поліморфна симуляція взаємодії частинок (Поліморфізм)	4
6	Магічні методи та створення класу «Вектор 3D»	4
7	Надійність фізичних розрахунків та Unit-тестування	4
8	Проектування гнучких систем за принципами SOLID	3
Семестр 3. Архітектурні шаблони, GUI та фінальний продукт		
9	Шаблони «Фабрика» та «Одинак» у всесвіті симуляцій	2
10	Створення першого вікна та інтерфейсу (GUI)	3
11	Інтерактивна модель «Маятник» з керуванням через GUI	3
12	Вбудовування динамічних графіків Matplotlib у GUI	3
13	Збереження експериментів (Серіалізація в JSON)	3
14	Документування коду за стандартами розробки	3
15	Пакування проєкту у готовий дистрибутив (Виконуваний файл)	3
16	Сервіси Google.	4
	Модульна контрольна робота	4
	Разом	58

6.4. Самостійна робота

№ з/п	Назва теми	Кількість годин
1	Аналітичний огляд методологій розробки. Порівняння Waterfall та Agile (Scrum/Kanban) для освітніх IT-проектів.	6
2	Вивчення Git-flow. Ознайомлення з розгалуженою системою ведення великих проектів та правилами написання зрозумілих комміт-повідомлень (Conventional Commits).	6
3	Ознайомлення з візуальним моделюванням архітектури програм. Побудова базових UML-діаграм класів (класи, атрибути, методи) для фізичних систем..	7
4	Глибоке вивчення механізму Name Mangling (спотворення імен) у Python та робота з дескрипторами даних для суворої валідації.	7
5	Дослідження проблеми множинного успадкування в Python. Вивчення алгоритму C3-лінеаризації (порядок роздільної здатності методів — MRO).	6
6	Порівняльний аналіз статичного поліморфізму (перевантаження методів) у мовах C++/Java та динамічного поліморфізму (Duck Typing) в Python.	7
7	Реалізація методів порівняння об'єктів (<code>__eq__</code> , <code>__lt__</code> , <code>__gt__</code>) та методів хешування (<code>__hash__</code>) для математичних та астрономічних об'єктів.	7
8	Ознайомлення з концепцією розробки через тестування (TDD — Test-Driven Development). Написання тест-кейсів перед написанням коду фізичних формул.	7
9	Детальний розбір та конспектування кожної літери SOLID. Аналіз антипатернів (помилки проектування) у великих обчислювальних програмах.	6
10	Ознайомлення з іншими важливими патернами: «Спостерігач» (Observer) — для відслідковування змін у фізичній системі, та «Стратегія» (Strategy) — для зміни алгоритмів розрахунку на льоту.	6
11	Створення архітектурних шаблонів для розробки плагінів або розширень до навчальних фізичних симуляторів.	6
12	Вивчення патерну архітектури інтерфейсів MVC (Model-View-Controller) або MVP. Розділення візуальної частини вікна від фізичних обчислень.	6
13	Оптимізація графічного виведення. Вивчення методів швидкої анімації графіків (FuncAnimation у Matplotlib) для плавного відображення руху тіл.	5
14	Ознайомлення з об'єктно-реляційним відображенням (ORM) на прикладі бібліотеки SQLAlchemy для збереження складних наукових даних.	5
15	Написання повноцінного інструментарію самодокументації за допомогою генератора Sphinx. Створення HTML-довідника користувача для своєї програми.	6
16	Дослідження сучасних методів ізоляції середовищ. Ознайомлення з концепцією контейнеризації (Docker) та пакуванням кросплатформних додатків.	6
	Разом	99

7. ІНСТРУМЕНТИ, ОБЛАДНАННЯ ТА ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ, ВИКОРИСТАННЯ ЯКИХ ПЕРЕДБАЧАЄ НАВЧАЛЬНА ДИСЦИПЛІНА

Технічні засоби: мультимедійний проектор.

Обладнання: персональні комп'ютери, пакет Microsoft Office, доступ в Інтернет.
Дистанційна платформа Moodle.

8. РЕКОМЕНДОВАНІ ДЖЕРЕЛА ІНФОРМАЦІЇ

Основна література

1. Васильєв О. М. Програмування в Python. Теорія і практика : навч. посіб. — Київ : Видавництво Ліра-К, 2023. — 462 с.
2. Бородкіна І. Л., Бородкін Г. О. Інженерія програмного забезпечення: Посібник для студентів вищих навчальних закладів. — Київ : Центр навчальної літератури, 2024. — 204 с.
3. Костюченко А. О. Об'єктно-орієнтоване програмування мовою PYTHON: Навчальний посібник. — Київ : КПІ ім. Ігоря Сікорського, 2021. — 112 с.
4. Ковтанюк М. С. Моделювання фізичних процесів мовою програмування Python з допомогою графічного модуля VPython: Навчально-методичний посібник. — Умань : УДПУ, 2021. — 96 с.
5. Роберт Мартін. Чистий код (Clean Code). — Київ : Видавництво "Фабула", 2019. — 448 с.

Допоміжна література

1. Dusty Phillips. Python 3 Object-Oriented Programming: Build robust and maintainable software with object-oriented design patterns. — 4th ed. — Birmingham : Packt Publishing,
2. Eric Freeman, Elisabeth Robson. Head First Design Patterns: Building Extensible and Maintainable Object-Oriented Software. — 2nd ed. — Sebastopol : O'Reilly Media, 2020. — 672 p.
3. Alexandre Neto. Physically Based Simulation in Python: An Object-Oriented Approach. — New York : Springer, 2023. — 280 p.