

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**
Навчально-науковий інститут високих технологій

Кафедра нанofізики конденсованих середовищ



«ЗАТВЕРДЖУЮ»

Заступник директора
з навчальної роботи

Грабчук Г.П.

«24» травня 2022 року

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Основи інженерії програмного забезпечення

для студентів

галузь знань 10 «Природничі науки»
напрямок 105 «Прикладна фізика»
освітній рівень бакалавр
освітня програма «Нанofізика та комп'ютерні технології»
вид дисципліни за вибором

Форма навчання	денна
Навчальний рік	2022/2023
Семестр	7
Кількість кредитів ECTS	3
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	

Сусь Богдан Богданович, к. ф.-м. н., асистент, кафедра нанofізики конденсованих середовищ

Пролонговано: на 20__/20__ н.р. _____ (_____) «__» 20__ р.
(підпис, ПІБ, дата)

на 20__/20__ н.р. _____ (_____) «__» 20__ р.
(підпис, ПІБ, дата)

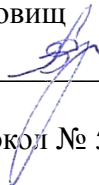
КИЇВ – 2022

Розробники:

Сусь Богдан Богданович, к. ф.-м. н., асистент, кафедра нанофізики конденсованих середовищ

ЗАТВЕРДЖЕНО

Зав. кафедри нанофізики конденсованих середовищ



Валерій Скришевський

Протокол № 5 від «19» квітня 2022 р.

Схвалено науково-методичною комісією інституту високих технологій

Протокол від «13» травня 2022 року № 4

Голова науково-методичної комісії



Русінчук Н. М.

ВСТУП

1. Мета дисципліни – ознайомлення студентів з відомими сучасними методами розробки програмних середовищ. Курс „Інженерія програмного забезпечення” є важливою складовою підвищення фундаментальної підготовки студентів та вдосконалення їх умінь на старших курсах.

2. Попередні вимоги до опанування або вибору навчальної дисципліни (за наявності):

1. Знати основні поняття в області програмування та теорії алгоритмів.
2. Вміти використовувати клієнт-серверні технології, бази даних та програми на базовому рівні.
3. Володіти елементарними навичками складання алгоритмів та написання програм для реалізації обчислень.

3. Анотація навчальної дисципліни:

Предметом навчальної дисципліни „Інженерія програмного забезпечення” є моделі розробки програмного забезпечення.

Навчальна дисципліна є складовою освітньо-професійної програми підготовки фахівців за першим (бакалаврським) рівнем вищої освіти галузі знань 10 „Природничі науки” зі спеціальності 105 „Прикладна фізика”. В курсі детально розглядаються основні етапи розробки програмних систем, сучасні гнучкі методології розробки програмного забезпечення найбільш поширені підходи до проведення моделювання та тестування сучасних інформаційних систем.

4. Завдання (навчальні цілі):

Дисципліна спрямована на формування програмних компетентностей:

- ЗК1. Здатність застосовувати знання у практичних ситуаціях.
- ЗК2. Знання та розуміння предметної області та розуміння професійної діяльності.
- ЗК3. Здатність застосовувати знання у практичних ситуаціях.
- ЗК5. Навички використання інформаційних і комунікаційних технологій.
- ЗК7. Вміння виявляти, ставити та вирішувати проблеми.
- ЗК9 Здатність працювати автономно.
- ЗК14 Здатність бути критичним і самокритичним.
- ЗК16 Здатність оцінювати та забезпечувати якість виконуваних робіт.
- ЗК17 Здатність спілкуватися з представниками інших професійних груп різного рівня (з експертами з інших галузей знань/видів економічної діяльності).
- ЗК18. Здатність працювати в команді.

ФК1. Здатність брати участь у плануванні та виконанні наукових та науково-технічних проектів.

ФК5 Здатність до постійного розвитку компетентностей у сфері прикладної фізики, інженерії та комп’ютерних технологій.

ФК8 Здатність працювати в колективах виконавців, у тому числі в міждисциплінарних проектах.

ФК10 Здатність реалізовувати автоматизацію експериментальних досліджень у різних сферах науки із використанням сучасних комп’ютерних технологій.

ФК11 Здатність використовувати комп’ютерні технології при проектуванні, розробці та діагностиці електронного обладнання.

ФК12-2 Вибірковий блок 2 Знання фізичних основ сучасного експериментального обладнання та вміння застосовувати їх до вибору, проектування, виготовлення та удосконалення вимірювальних приладів для застосувань у природничих науках.

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			
1.1	Знати основні підходи до створення програмного забезпечення, розуміти методи для розроблення алгоритмічного програмного забезпечення. Знати основні принципи архітектурного та об'єктно-орієнтованого проектування.	Лекції	Письмова контрольна робота Запитання на заліку	20% 12%
2.1	Вміти застосовувати спеціальне програмне забезпечення і системи автоматизації проектування для створення комп'ютерних моделей реальних об'єктів, систем, мереж та інтернет додатків. Застосовувати набуті знання з інженерії програмного забезпечення при розв'язанні практичних задач.	Лабораторні роботи	Звіти по лабораторних роботах Задача на заліку	20% 12%
4.1	Прийняти і обґрунтувати рішення щодо вибору типу моделі, підходів моделювання програмного комплексу та тестування програмного забезпечення. Ефективно використовувати сучасні методи та інструментарій для розробки програмного забезпечення.	Лекції, лабораторні роботи, самостійна робота студента	Письмовий звіт з семестрової роботи Практичне завдання на заліку	20% 18%

6. Співвідношення результатів навчання дисципліни із програмними результатами навчання

Результати навчання дисципліни					
	1.1	2.1	3.1	3.2	4.1
Програмні результати навчання (назва)					
ПРН 4 Застосовувати фізичні, математичні та комп'ютерні моделі для дослідження фізичних явищ, розробки приладів і наукоємних технологій.	+			+	
ПРН 6 Відшукувати необхідну науково-технічну інформацію в науковій літературі, електронних базах, інших джерелах, оцінювати надійність та релевантність інформації.			+	+	
ПРН 8 Вільно спілкуватися з професійних питань державною та англійською мовами усно та письмово.			+	+	+
ПРН 9 Презентувати результати досліджень і розробок фахівцям і нефахівцям, аргументувати власну позицію.				+	+
ПРН 10 Планувати й організовувати результативну професійну діяльність індивідуально і як член команди при розробці та реалізації наукових і прикладних проектів.	+		+		
ПРН 11 Знати цілі сталого розвитку та можливості своєї професійної сфери для їх досягнення, в тому числі в Україні.			+	+	+
ПРН 12 Розуміти закономірності розвитку прикладної фізики, її місце в розвитку техніки, технологій і суспільства, у тому числі в розв'язанні екологічних проблем.			+	+	+

ПРН 14 Організувати результативну роботу індивідуально і як член команди.	+	+		+	+
ПРН 15 Розробляти та формулювати свої професійні висновки та розумно їх аргументувати для фахової та нефахової аудиторії.	+		+		+
ПРН 16 Оцінювати важливість матеріалів для досягнення цілей наукового дослідження в галузі прикладної фізики зі спеціалізацією в сфері нанофізики.	+				
ПРН 17 Представляти і захищати отримані наукові і практичні результати в усній та письмовій формі. ПРН 17	+		+	+	+
ПРН 18 Використовувати сучасні комп'ютерні технології при розробці, виготовленні, діагностиці та автоматизації обладнання.	+	+		+	
ПРН19-1 Вибірковий блок 1: На основі отриманих знань проектувати електронні прилади та програмне забезпечення для потреб нанотехнологій.	+	+		+	
ПРН19-2 Вибірковий блок 2: На основі отриманих знань проектувати та створювати автоматизовані експериментальні установки для проведення досліджень в природничих науках.	+	+		+	

7. Схема формування оцінки.

7.1 Форми оцінювання студентів:

- семестрове оцінювання:

1. Контрольна робота: РН 1.1 - 30 балів/18 балів.
 2. Звіти по лабораторних роботах: РН 2.1. - 30 балів/18 балів.
 3. Самостійна семестрова робота: РН 4.1 - 40 балів/24 балів.
- Усього: 100 балів/60 балів.

- підсумкове оцінювання:

- Письмовий залік: 1 теоретичне запитання (12 балів/8 балів, оцінює РН 1.1), 1 задача (12 балів/8 балів, оцінює РН 2.1), 1 практичне завдання (16 балів/8 балів, оцінює РН 4.1);
- Максимальна кількість балів які можуть бути отримані студентом - 40 балів;
- Для отримання загальної позитивної оцінки з дисципліни оцінка за залік не може бути меншою 24 балів;
- Студент не допускається до заліку, якщо під час семестру набрав менше ніж 36 балів.
- Студент допускається до заліку за умови виконання всіх передбачених планом лабораторних робіт.
- Студент допускається до заліку за умови виконання самостійної семестрової роботи.

Оцінювання	Min	Max
Семестрове оцінювання	36	60
Підсумкове оцінювання	24	40
Всього	60	100

7.2 Організація оцінювання:

У кінці семестру після завершення вивчення тем на останній лекції проводиться письмова контрольна робота. Контрольна робота спрямована на визначення рівня знань студентами матеріалів лекцій за весь семестр.

Протягом семестру студенти виконують лабораторні роботи, за результатами чого готують письмові та усні звіти.

Протягом семестру студенти працюють над виконанням самостійної роботи, необхідні знання та навички для виконання якої отримують під час лекційних та лабораторних занять. Результатом виконання семестрового завдання є письмовий звіт та усний захист.

Для студентів, які упродовж семестру не досягли мінімального рубіжного рівня оцінки (36 балів), для одержання допуску до іспиту обов'язковим є виконання додаткових завдань.

Під час заліку студенти мають надати відповідь на одне теоретичне запитання, розв'язати одну задачу та виконати практичне завдання.

7.3 Шкала відповідності оцінок

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

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

№ п/п	Назва теми	Кількість годин		
		лекції	лабораторні	самостійна робота
Частина 1 Теоретичне навчання				
1	Вступ. Тема 1 Вступ. Мета та завдання дисципліни. Основні поняття інженерії програмного забезпечення. Загальні підходи створення програмного забезпечення. Огляд етапів розробки програмного забезпечення. Програмне забезпечення автоматизованих систем.	2		2
2	Тема 2. Основні принципи проектування програм. Передумови появи гнучких методологій розробки програмного забезпечення. Текст Agile маніфесту та його основні принципи у порівнянні з класичною методологією розробки.	2		2
3	Тема 3. Методи, моделі та технології гнучкої розробки програмного забезпечення. Основні моделі розробки програмного забезпечення: каскадна, спіральна, інкрементна та ітераційна моделі.	2		2
4	Тема 4. Процедурне, компонентно-орієнтоване, функціональне та системне програмування. Багатопроцесорні системи та потоки. Формування та специфікація вимог до програмного забезпечення.	2		2
5	Тема 5. Взаємодія програмних додатків. Архітектура .NET. Microsoft Visual Studio IDE. Основні концепти C# та Common Language Runtime.	2		2
6	Тема 6. Клієнт-серверна взаємодія. Системи контролю версій.	2		2
7	Тема 7. Методологія функціонального моделювання IDEF0. Побудова моделей IDEF0.	2		2
8	Тема 8. UML: історія та специфікації. Концептуальна модель мови UML. діаграми UML статичного та динамічного подання об'єктно-орієнтованих програмних систем.	2		2
9	Тема 9. Дослідження діаграми варіантів використання. Об'єктно-орієнтований аналіз та побудова специфікацій.	2		2
10	Тема 10. Контрольна робота.	1		2
11	Тема 11. WEB. PHP	2		2
12	Тема 12 Поняття платформи програмного забезпечення та кросплатформності. Способи розробки кросплатформного програмного забезпечення. Java-платформа	2		2
13	Тема 13. Віртуальна машина java. Just-in-time компіляція: переваги та недоліки. Алгоритмічні структури. Види алгоритмів.	2		2
14	Тема 14. Засоби програмної інженерії в наукових дослідженнях. Системи автоматизованої обробки інформації. Програмування потоків даних.	3		2
Частина 2 Лабораторний практикум				
23	Тема 15. Visual Studio .NET. Створення, виконання та компілювання програм. Проектування інтерфейсу.		2	2
24	Тема 16. Система JIRA. Робочі процеси. Робочий простір користувача.		2	2
25	Тема 17. Основні принципи C #, CLR. Створення консольного застосування. Компіляція та виконання. Відлагодження.		2	2
26	Тема 18. Операції читання та записування даних.		2	2

	<i>Читання та записування об'єктів в файл</i>			
27	Тема 19. Використання систем контролю версій вихідного коду програм. <i>Git.</i>		2	2
28	Тема 20. Графічний інтерфейс користувача. <i>Особливості розробки.</i>		2	2
29	Тема 21. <i>Діаграми класів на мові UML</i>		2	2
30	Тема 22. Використання засобів автоматизації тестування програмного забезпечення.		2	2
31	Тема 23. Рекурсивні та не рекурсивні методи.		2	2
32	Тема 24. <i>Python. Типи даних Модульні програми.</i>		2	2
33	Тема 25. <i>Робота з периферійними пристроями та LabView.</i>		4	
	ВСЬОГО	28	14	48

Загальний обсяг 90 год., в тому числі:

Лекцій – **28 год.**

Лабораторні заняття - **14 год.**

Самостійна робота - **48 год.**

9. Рекомендовані джерела:

Основна:

1. C# for Experienced Programmers (Deitel Developer Series) by Harvey M. Deitel. Pearson Education. 2006. 1456 p.
2. C# 6.0 and the .NET 4.6 Framework. Andrew Troelsen, Philip Japikse.
3. https://www.amazon.com/C-6-0-NET-4-6-Framework/dp/1484213335/ref=sr_1_1?s=books&ie=UTF8&qid=1496819049&sr=1-1&keywords=.net
4. Professional C# 6 and .NET Core 1.0. Christian Nagel.
5. Sommerville, Ian Software engineering / Ian Sommerville. — 9th ed. Pearson. 790 p.
6. CLR via C# (4th Edition). Jeffrey Richter.
<https://www.amazon.com/CLR-via-4th-Developer-Reference/dp/0735667454>
7. <http://refcardz.dzone.com/refcardz/junit-and-easymock>
8. <http://junit.sourceforge.net/doc/testinfected/testing.htm>
9. <http://junit.sourceforge.net/doc/faq/faq.htm>
10. <http://junit.sourceforge.net/javadoc/index.html>
11. <http://www.javaworld.com/javaworld/jw-12-2000/jw-1221-junit.html>
12. <http://www.ibm.com/developerworks/ru/edu/jjunit4/index.html>
13. <http://litvinyuk.com/articles/junit.htm>

10. Додаткові ресурси: немає.