

КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
ІМЕНІ ТАРАСА ШЕВЧЕНКА

Інститут високих технологій

Кафедра нанofізики конденсованих середовищ



**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Інженерія програмного забезпечення

(повна назва дисципліни)

для студентів

галузь знань 09 Біологія  
(шифр і назва)  
спеціальність 091 Біологія  
(шифр і назва спеціальності)  
освітній рівень магістр  
(молодший бакалавр, бакалавр, магістр)  
освітня програма Біоінформатика та структурна біологія  
(назва освітньої програми)  
вид дисципліни обов'язкова

Форма навчання	денна
Навчальний рік	2020/2021
Семестр	2
Кількість кредитів ECTS	4
Мова викладання, навчання та оцінювання	українська
Форма заключного контролю	екзамен

Викладач: Сусь Богдан Богданович, асистент кафедри нанofізики конденсованих середовищ

Пролонговано: на 20 20 / 20 22 н.р. Григорук (А.М. Григорук) «05» 03 2021 р. прот. 23  
(підпис, ПІБ, дата)

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

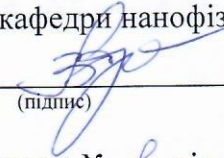
КИЇВ – 2020

Розробники:

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

ЗАТВЕРДЖЕНО

Зав. кафедри нанофізики конденсованих середовищ

  
\_\_\_\_\_

(Скришевський В.А.)  
(прізвище та ініціали)

Протокол № 8 від « 12 » червня 2020 р.

Схвалено науково - методичною комісією Інституту високих технологій

Протокол від « 24 » червня 2020 року № 3

Голова науково-методичної комісії

  
\_\_\_\_\_

(Русінчук Н.М.)  
(прізвище та ініціали)

## 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 другим (магістерським) рівнем вищої освіти галузі знань 09 „Біологія” зі спеціальності 091 „Біологія”. 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):**

Навчальні цілі дисципліни спрямовані на досягнення таких загальних та фахових компетентностей:  
ЗК02. Здатність використовувати інформаційні та комунікаційні технології.

СК01. Здатність користуватися новітніми досягненнями біології, необхідними для професійної, дослідницької та/або інноваційної діяльності. Здатність застосовувати знання у професійній діяльності з урахуванням новітніх досягнень, у т.ч. для дослідницької роботи.

СК02. Здатність формулювати задачі моделювання, створювати моделі об'єктів і процесів на прикладі різних рівнів організації живого із використанням математичних методів і інформаційних технологій.

СК03. Здатність користуватися сучасними інформаційними технологіями та аналізувати інформацію в галузі біології і на межі предметних галузей.

СК11. Вміння розробляти програмне забезпечення для обробки біомолекулярних даних.

**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.	Лекції	Письмова контрольна робота	20%
1.2.	Знати основні принципи архітектурного та об'єктно-орієнтованого проектування.	Лекції	Запитання на екзамені	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	Лабораторні роботи	Звіти по лабораторних роботах	20%
2.2.	Застосовувати набуті знання з інженерії програмного забезпечення при розв'язанні практичних задач.	Лабораторні роботи	Задача на екзамені	12%
4.1	Прийняти і обґрунтувати рішення щодо	Лекції, лабораторні	Письмовий звіт з	20%

	<i>вибору типу моделі, підходів моделювання програмного комплексу та тестування програмного забезпечення. Ефективно використовувати сучасні методи та інструментарій для розробки програмного забезпечення.</i>	<i>роботи, самостійна робота студента</i>	<i>семестрової роботи</i>  <i>Практичне завдання на екзамені</i>	16%
--	---	---	--	-----

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

<b>Результати навчання дисципліни</b> <b>Програмні результати навчання</b>	1.1	1.2	2.1	2.2	4.1
ПРН2. Використовувати бібліотеки, інформаційні бази даних, інтернет ресурси для пошуку необхідної інформації.	+				+
ПРН3. Знаходити шляхи швидкого і ефективного розв'язку поставленого завдання, генерування ідей, використовуючи отримані знання та навички.	+	+	+		
ПРН10. Вміти моделювати основні процеси дослідження з метою вибору методів дослідження, апаратного забезпечення або створення нових методик.			+		+
ПРН11. Вміти проводити статистичну обробку, аналіз та узагальнення отриманих експериментальних даних із використанням програмних засобів та сучасних інформаційних технологій, що використовують в галузі біології.			+		+
ПРН14. Використовувати інноваційні підходи для розв'язання конкретних біологічних завдань.	+	+	+	+	+
ПРН17. Моделювати об'єкти і процеси у живих організмах та їхніх компонентах із використанням математичних методів й інформаційних технологій.	+	+	+	+	+

## 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 екзамен: 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 екзамен не може бути меншою 24 балів;
- Студент не допускається до екзамену, якщо під час семестру набрав менше ніж 20 балів (рекомендований мінімум 36 балів).
- Студент допускається до екзамену за умови виконання всіх передбачених планом лабораторних робіт.
- Студент допускається до екзамену за умови виконання самостійної семестрової роботи.

Оцінювання	Min	Max
Семестрове оцінювання	36	60
Екзамен	24	40
<b>Всього</b>	<b>60</b>	<b>100</b>

### 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 екзамену студенти мають надати відповідь на одне теоретичне запитання, розв'язати одну задачу та виконати практичне завдання.

### 7.3 Шкала відповідності оцінок

Відмінно / Excellent	90-100
Добре / Good	75-89
Задовільно / Satisfactory	60-74
Незадовільно / Fail	0-59
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
<b>Частина 1 Теоретичне навчання</b>				
1	<b>Вступ.</b> <b>Тема 1</b> Вступ. Мета та завдання дисципліни. Основні поняття інженерії програмного забезпечення. Загальні підходи створення програмного забезпечення. Огляд етапів розробки програмного забезпечення. Програмне забезпечення автоматизованих систем.	1		1
2	<b>Тема 2.</b> Основні принципи проектування програм. Передумови появи гнучких методологій розробки програмного забезпечення. Текст Agile маніфесту та його основні принципи у порівнянні з класичною методологією розробки.	1		1
3	<b>Тема 3.</b> Методи, моделі та технології гнучкої розробки програмного забезпечення. Основні моделі розробки програмного забезпечення: каскадна, спіральна, інкрементна та ітераційна моделі.	1		1
4	<b>Тема 4.</b> Основні характеристики методології Scrum. Ролі персоналу у проекті та артефакти. Особливості застосування методології до розробки програмних проектів.	1		1
5	<b>Тема 5.</b> Процедурне, компонентно-орієнтоване, функціональне та системне програмування. Багатопроцесорні системи та потоки. Формування та специфікація вимог до програмного забезпечення.	1		1
6	<b>Тема 6.</b> Взаємодія програмних додатків. Архітектура .NET . Microsoft Visual Studio IDE. Основні концепти C# та Common Language Runtime.	1		1
7	<b>Тема 7.</b> Клієнт-серверна взаємодія. Системи контролю версій.	1		1
8	<b>Тема 8.</b> Поняття основних процесів керування проектом. Основні поняття методології проектування інформаційних систем. Організація проектної команди. Стратегії керівництва. Становлення команди.	1		1
9	<b>Тема 9.</b> Методологія функціонального моделювання IDEF0. Побудова моделей IDEF0.	1		1
10	<b>Тема 10.</b> Методологія опису технологічних процесів IDEF3. Синтаксис та семантика діаграм DFD.	1		1
11	<b>Тема 11.</b> UML: історія та специфікації. Концептуальна модель мови UML. діаграми UML статичного та динамічного подання об'єктно-орієнтованих програмних систем.	1		1
12	<b>Тема 12.</b> Дослідження діаграми варіантів використання. Об'єктно-орієнтований аналіз та побудова специфікацій.	1		1
13	<b>Тема 13.</b> Контрольна робота.	1		1
14	<b>Тема 14.</b> WEB. PHP	1		1
15	<b>Тема 15</b> Поняття платформи програмного забезпечення та кросплатформності. Способи розробки кросплатформного програмного забезпечення. Java-платформа	1		1
16	<b>Тема 16.</b> Віртуальна машина java. Just-in-time компіляція: переваги та недоліки.	1		1
17	<b>Тема 17.</b> Способи передачі даних між системами. JSON. Мова керування даними. Бази даних. SQL.	1		1
18	<b>Тема 18.</b> Якість та тестування програмних продуктів. Методи верифікації, валідації та тестування програмного забезпечення. Типи тестів та планування	1		1

	<i>тестів. Виведення помилок. Особливості тестування веб-додатків, експлуатації та супроводу програмних систем.</i>			
	<b>Тема 19.</b> Алгоритмічні структури у мові Python. Види алгоритмів.	1		1
	<b>Тема 20.</b> Програмне забезпечення для організації веб-сайтів. CMS. Системи керування вмістом.	1		1
	<b>Тема 21.</b> Засоби програмної інженерії в наукових дослідженнях та навчанні. Системи автоматизованої обробки інформації. Програмування потоків даних. Графічна мова програмування "G". Автоматизація використання обчислювального та вимірювального лабораторного обладнання. LabView та взаємодія з елементами, реалізованими на платформі Microsoft .NET. Особливості роботи з інтерфейсами та периферійним обладнанням.			10
	<b>Тема 22.</b> Особливості створення віртуальних лабораторних робіт та віртуальних симуляторів. Багатоплатформові інструменти для розробки дво- та тривимірних додатків. Сценарії на C#. Система Unity.			10
<b>Частина 2 Лабораторний практикум</b>				
23	<b>Тема 23.</b> Visual Studio .NET. Створення, виконання та компілювання програм. Проектування інтерфейсу.		2	3
24	<b>Тема 24.</b> Система JIRA. Робочі процеси. Робочий простір користувача.		2	3
25	<b>Тема 25.</b> Основні принципи C #, CLR. Створення консольного застосування. Компіляція та виконання. Відлагодження.		2	4
26	<b>Тема 26.</b> Операції читання та записування даних. Читання та записування об'єктів в файл		2	3
27	<b>Тема 27.</b> Використання систем контролю версій вихідного коду програм. Git.		2	3
28	<b>Тема 28.</b> Графічний інтерфейс користувача. Особливості розробки.		2	3
29	<b>Тема 29.</b> Діаграми класів на мові UML		2	3
30	<b>Тема 30.</b> Використання засобів автоматизації тестування програмного забезпечення.		2	3
31	<b>Тема 31.</b> Рекурсивні та не рекурсивні методи.		2	3
32	<b>Тема 32.</b> Python. Типи даних Модульні програми.		1	6
33	<b>Тема 33.</b> Робота з периферійними пристроями та LabView.		1	6
	<b>ВСЬОГО</b>	<b>20</b>	<b>20</b>	<b>80</b>

**Загальний обсяг 120 год.,** в тому числі:

Лекцій – **20 год.**

Лабораторні заняття - **20 год.**

Самостійна робота - **80 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](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. <https://www.amazon.com/Professional-NET-Core-Christian->

Nagel/dp/111909660X/ref=sr\_1\_16?s=books&ie=UTF8&qid=1496819302&sr=1-16&keywords=.net

5. CLR via C# (4th Edition). Jeffrey Richter. <https://www.amazon.com/CLR-via-4th-Developer-Reference/dp/0735667454>
6. Ахо А., Лам М., Сети Р., Ульман Д. Компиляторы: принципы, технологии и инструментарий, 2-е изд. : Пер. с англ. - М. : 000 "И.Д. Вильямс", 2008. - 1184 с.
7. Волохов В., Системне програмування, Київ, КНУ, 2010.
8. Таненбаум С. Современные операционные системы, 2-е изд.: - СПб.: Питер, 2004. – 1040 с.
9. Столингс В. Операционные системы, 4-е изд.: - М.: Издательский дом "Вильямс", 2002., 848 с.
10. Предметно-ориентированное программирование. Структуризация сложных программных систем.: Пер. С англ. -М.: ООО «И. Д. Вильямс», 2011. 448 с.: ил. — Парал. тит. Англ., ISBN 978-5-8459-1597-9
11. Роберт Мартин, Мика Мартин. Принципы, паттерны и методики гибкой разработки на языке C# Пер. с англ. –СПб.: Символ-Плюс, 2011. –768 с.
12. В. В. Кулямин. Технологии программирования. Компонентный подход
13. Фаулер, Мартин. Архитектура корпоративных программных приложений.: Пер. с англ. —М.: Издательский дом "Вильямс", 2006. —544 с.
14. Орлов С.А. Технологии разработки программного обеспечения. Разработка сложных программных систем: учебник / Орлов С. А.; .-3-е изд.. -СПб.: ПИТЕР, 2004.-527 с.
15. М. Фаулер, К. Скотт. UML в кратком изложении. Применение стандартного языка объектного моделирования: Пер. с англ. –М.: Мир, 1999. –191 с
16. <http://refcardz.dzone.com/refcardz/junit-and-easymock>
17. <http://junit.sourceforge.net/doc/testinfected/testing.htm>
18. <http://junit.sourceforge.net/doc/faq/faq.htm>
19. <http://junit.sourceforge.net/javadoc/index.html>
20. <http://www.javaworld.com/javaworld/jw-12-2000/jw-1221-junit.html>
21. <http://www.ibm.com/developerworks/ru/edu/jjunit4/index.html>
22. <http://litvinyuk.com/articles/junit.htm>
23. Доусон М. Програмуємо на Python. - СПб.: Питер, 2014. - 416 с
24. Хахаев И.А. Практикум по алгоритмизации и программированию на Python. М.: Альт Линукс, 2011. — 126 с.

**10. Додаткові ресурси:** немає.