

Міністерство освіти і науки України
Український державний університет науки і технологій

Факультет Комп'ютерних технологій і систем
Кафедра Комп'ютерні інформаційні технології



ЗАТВЕРДЖУЮ

Перший проректор

професор

Анатолій РАДКЕВИЧ

20 22 р.

РОБОЧА ПРОГРАМА
навчальної дисципліни

ОСНОВИ ПРОГРАМНОЇ ІНЖЕНЕРІЇ

- Рівень вищої освіти - перший (бакалаврський)
Статус дисципліни - обов'язкова
Обсяг - 6 кредитів ECTS

Дисципліна є компонентною освітньої програми:

Шифр галузі	Код і назва спеціальності	Назва ОПІ	Код навчальної дисципліни
12 Інформаційні технології	121 Інженерія програмного забезпечення	Інженерія програмного забезпечення	ОК 21

Форма підсумкового контролю - залік

м. Дніпро – 2022

Розробник робочої програми, доцент

Олександра ГОРБОВА

Робочу програму затверджено на засіданні кафедри «Комп'ютерні інформаційні технології»
«02» 02 2022 р. протокол № 10

Зав. кафедри, к.т.н., доц.

Вадим ГОРЯЧКІН

Схвалено вченою радою факультету «Комп'ютерних технологій і систем»
«30» 06 2022 р. протокол № 5

Голова вченої ради

Володимир МАЛОВІЧКО

ПОГОДЖЕННЯ:

Робоча програма дисципліни відповідає нормативам навчального плану

Керівник навчального відділу

Володимир ПУЛЬШІНСЬКИЙ

«25» 07 2022 р.

Робоча програма дисципліни відповідає вимогам нормативно-методичних документів

Навчально-методичний відділ

Галина БЕЛЕЙЧИК

«25» 07 2022 р.

1 Загальні відомості

Предметом дисципліни «Основи програмної інженерії» (ОК 21) є вивчення принципів та методів розробки та налагодження програмного забезпечення.

Метою дисципліни є досягнення компетентностей, які основані на зазначених в освітньо-професійній програмі (ОП).

1. ЗК-1. Здатність до абстрактного мислення, аналізу та синтезу.
2. ЗК-2. Здатність застосовувати знання у практичних ситуаціях.
3. ЗК-5. Здатність вчитися і оволодівати сучасними знаннями.
4. ЗК-6. Здатність до пошуку, оброблення та аналізу інформації з різних джерел.
5. ФК-1. Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення.
6. ФК-2. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.
7. ФК-3. Здатність розробляти архітектури, модулі та компоненти програмних систем.
8. ФК-8. Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення.
9. ФК-11. Здатність реалізувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.
10. ФК-14. Здатність до алгоритмічного та логічного мислення.

У процесі вивчення дисципліни у студентів формуються наступні **«соціальні навички»**.

Soft skills:

1. **Особистісні:** обирати вірні математичні рішення (**ОН2**) та розробки на них вірних алгоритмів (**ОН4**), розробляти вірні програмні коди (**ОН3**), тестувати та налагоджувати програми, розробляти технічну документацію на власну програму (**ОН4**) та (**ОН1**).
2. **Комунікаційні:** зрозуміло формулювати думки при обговоренні обраних методів та алгоритмів (**КН1**), розроблених за ними, аргументувати обрані типи та структури даних у програмі (**КН3**), вільно пропонувати методи розробки програмного коду (**КН2**), аргументовано доводити правильність роботи власної програми (**КН3**).
3. **Управлінські:** працювати в команді при обрані математичних методів на базі яких розробляється алгоритм (**УН1**), запобігати конфліктам при виборі алгоритмів розробки (**УН4**), тестуванні та налагодженні програми, доводити власну точку зору при виникненні неоднозначних ситуацій при написанні програмної документації одноосібно та працюючи у колектив (**УН2**).

2 Міждисциплінарні зв'язки

Перелік дисциплін які потрібні для вивчення дисципліни «Основи програмної інженерії»	
ОК 4	Вища математика (математичний аналіз)
ОК 5	Лінійна алгебра та аналітична геометрія
ОК 10	Основи програмування
ОК 12	Архітектура комп'ютера
Перелік дисциплін, вивчення яких спирається на дисципліну «Основи програмної інженерії»	
ОК 7	Алгоритми та структури даних
ОК 13	Аналіз вимог до програмного забезпечення
ОК 14	Архітектура та проектування програмних засобів
ОК 18	Об'єктно-орієнтоване програмування
ОК 27	Людино-машинна взаємодія

3 Очікувані результати навчання

Дисципліна Основи програмування повинна забезпечити такі результати навчання (згідно з ОП).

1.7 Програмні результати навчання (ПРН)

ПРН-3. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.

ПРН-6. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.

ПРН-7. Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.

ПРН-9. Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.

ПРН-11. Вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання.

ПРН-13. Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.

ПРН-14. Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення.

ПРН-19. Знати та вміти застосовувати методи верифікації та валідації програмного забезпечення

ПРН-23. Вміти документувати та презентувати результати розробки програмного забезпечення.

Очікуванні результати навчання (ОРН), які повинні бути досягнуті після опанування дисципліни «Основи програмної інженерії»

№	ОРН Рівень	Шифр	ПРН
1	Перераховувати основні етапи життєвого циклу програмного забезпечення	I	ПРН-3
2	Називати методи моделювання, налагодження та тестування програмного забезпечення	I	ПРН-7, ПРН-13
3	Вміти виконувати передпроектне обстеження предметної області	I	ПРН-9, ПРН-11
4	Знати основні методи розробки програмного забезпечення	I	ПРН-9, ПРН-11, ПРН-13, ПРН-19
5	Знати методи побудови архітектури програмних систем.	II	ПРН-3, ПРН-6
6	Перераховувати вимоги до супроводження програмного забезпечення протягом всього життєвого циклу.	II	ПРН-3, ПРН-10, ПРН-12
7	Створювати алгоритми до програм	II	ПРН-13
7	Створювати програми із застосуванням математичних знань	II	ПРН-7, ПРН-13
8	Створювати найпростіші програми із застосуванням базових конструкцій	III	ПРН-7, ПРН-13
9	Застосовувати підходи щодо проектування програмного забезпечення	III	ПРН-3, ПРН-7, ПРН-13
10	Застосовувати різні парадигми програмування при програмуванні	III	ПРН-7, ПРН-11
11	Застосовувати основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення.	III	ПРН-7, ПРН-3, ПРН-19
12	Застосовувати найпростіших алгоритмів при написанні своїх програм	III	ПРН-6, ПРН-11, ПРН-13
	Застосовувати сучасні методи розробки програмного забезпечення	III	ПРН-7, ПРН-13
13	Аналізувати правильність написаного алгоритму	IV	ПРН-14
14	Аналізувати правильність застосування методів, що були використані під час створення програмного забезпечення.	IV	ПРН-13
15	Документувати програму та виконану роботу	IV	ПРН-14, ПРН-23
16	Вміти поділити програму на самостійні частини (модулі)	IV	ПРН-6

4 Критерії оцінювання результатів навчання

Шкала ЕКТС	Очікуванні результати навчання
A	Вміє застосовувати конструкції та оператори мови C++, виявляє творчий підхід і правильно обґрунтовує прийняті рішення щодо вибору алгоритму програмування, добре володіє різносторонніми вміннями та навичками при написанні програм. Вірно використовує методи розробки програмного забезпечення, створює структури даних у програмі та застосовує їх у своїх програмах, під час всього життєвого циклу програмного забезпечення. Правильно документує програму та процес розробки програмного забезпечення.
B	Вміє застосовувати конструкції та оператори мови C++, виявляє творчий підхід і правильно обґрунтовує прийняті рішення по вибору алгоритму програмування, добре володіє різносторонніми вміннями та навичками при написанні програм, проте його програми містять певні недоліки. Правильно документує програму та процес розробки програмного забезпечення.
C	Вміє застосовувати конструкції та оператори мови C++, знає методи розробки програмного забезпечення, проте не вміє вірно застосувати їх при написанні алгоритмів та програм. Документує програму та процес розробки програмного забезпечення.
D	Вміє створити алгоритм, написати за ним програму. Знає основні етапи життєвого циклу програмного забезпечення. Розробка програм ведеться не структуровано та непослідовно. Вміти виконувати процес розробки програмного забезпечення.
E	Знає методи розробки програмного забезпечення. Вміє створити алгоритм та написати за ним програму. Вміти документувати програму
FX	Знає основні етапи життєвого циклу програмного забезпечення, припускає принципові помилки при написанні алгоритмів та програм, невірно застосовує основні оператори та конструкції мови C++.
F	Знає основні етапи життєвого циклу програмних систем та методи розробки програмного забезпечення.

Досягнення вищих оцінок за шкалою ЕКТС базується на досягнутих нижчих.

5 Види діагностування результатів навчання

Вид контролю	Бал
ПК1	50
ПК2	50

Співставлення шкал оцінювання

Екзамен, диференційований залік, курсова робота/проект

	Шкала ЕКТС	Оцінка за чотирибальною шкалою	
90-100	A	Відмінно	відмінно
82-89	B	Добре	дуже добре
75-81	C		добре
67-74	D	Задовільно	задовільно
60-66	E		достатньо
35-59	FX	Незадовільно	незадовільно з повторним складанням контрольного заходу
1-34	F		незадовільно з повторним вивченням дисципліни

6 Розподіл навчального часу для денної форми навчання

Вид навчання	Семестр		Усього	
	другий			
	I половина	II половина		
	год	год	год	кр. ECTS
Загальний обсяг за навчальним планом	90	90	180	6
Навчальні заняття:	40	40	80	
– лекції	16	16	32	
– лабораторні заняття	16	16	32	
– практичні заняття	8	8	16	
Самостійна робота:	50	50	100	
– підготовка до навчальних занять	20	20	40	
– опрацювання розділів програми, які не викладаються на лекціях	8	7	15	
– підготовка до контрольних заходів	22	23	45	
– підсумковий контроль				залік

7 Зміст дисципліни

Тема лекції (заняття)	Обсяг годин	СН
Модуль 1.		
Лекції		
1. Основи життєвого циклу програмних засобів	4	
2. Покрокова розробка програм	2	
3. Відлагодження програм	2	
4. Тестування програмних систем	6	
5. Процес розробки ПЗ	2	
Практичні заняття		
1. Розробка алгоритму методом покрокової деталізації.	2	ОН2, ОН4, КН1, УН1, УН4, УН2
2. Модулі. Механізми передачі параметрів	2	ОН2, ОН4, КН1, УН1, УН4, УН2
3. Тестування коду. Розробка автотестів.	2	ОН2, ОН4, КН1, УН1, УН4, УН2
4. Розробка алгоритмів, написання програм та їх тестування. Розбиття програми на модулі	2	ОН2, ОН4, КН1, УН1, УН4, УН2
Лабораторні роботи		
1. Модульне програмування. Розробка програми методом низхідного програмування.	6	ОН1, ОН2, ОН3, ОН4, КН2, КН3, КН4, УН1
2. Модульне програмування. Метод висхідного програмування.	4	ОН1, ОН2, ОН3, ОН4, КН2, КН3, КН4, УН1
3. Налаштування програм.	6	ОН4, КН2, КН3, КН4, УН1
Самостійна робота		
1. Підготовка до навчальних занять	20	
2. Поглиблене вивчення конструкцій та операторів С++	8	
3. Підготовка до контрольних заходів	22	
Усього	90/3	
Модуль 2		
Лекції		
6. Розробка інтерфейсу користувача	4	
7. Встановлення вимог	4	
8. Розробка інформаційної моделі	2	
9. Інженерія вимог Джекобсон. Модель сценаріїв	2	
10. Розробка програмної документації	2	
11. Методи керування програмним проектом	2	
Практичні заняття		
5. Створення застосунку Windows Forms	2	ОН2, ОН4, КН1, УН1, УН4, УН2
6. Створення застосунку Windows Forms для обробки матриці. Робота з таблицею.	2	ОН2, ОН4, КН1, УН1, УН4, УН2
7. Основні конструкції мови Python. Написання простих програм	2	ОН2, ОН4, КН1, УН1, УН4, УН2

8. Написання програм мовою Python. (не тривіальні випадки). Підсумкове заняття	2	ОН2, ОН4, КН1, УН1, УН4, УН2
Лабораторні роботи		
4. Розробка програм із графічним інтерфейсом користувача.	6	ОН1, ОН2, ОН3, ОН4, КН2, КН3, КН4, УН1
5. Розробка простих програм засобами мови Python.	4	ОН1, ОН2, ОН3, ОН4, КН2, КН3, КН4, УН1
6. Використання змінюваних типів даних для реалізації рекомендаційної системи	6	ОН1, ОН2, ОН3, ОН4, КН2, КН3, КН4, УН1
Самостійна робота		
1. Підготовка до навчальних занять	20	
2. Поглиблене вивчення Visual Studio у частині розробки віконного інтерфейсу	3	
3. Поглиблене вивчення використання основних конструкцій мови Python. Бібліотеки мови.	4	
4. Підготовка до контрольних заходів	23	
Усього	90/3	

8 Складова робочої програми дисципліни для заочної форми навчання

Дисципліна - Основи програмної інженерії

Кафедра Комп'ютерних інформаційних технологій

Код і назва спеціальності	Назва ОП	Код навчальної дисципліни
121 Інженерія програмного забезпечення	Інженерія програмного забезпечення	ОК 21

Розподіл навчального часу за навчальним планом

Курс навчання	№ семестру	Навантаження у семестрі, год/кредит	Аудиторні заняття				Самостійна робота, год	Контрольна робота, одиниці	Форма підсумкового контролю
			Всього	У тому числі					
				лекції	лабораторні	практичні			
1	2	180/6	12	6	6	-	168	-	Залік

Календарний план навчальних занять і робіт

Номер семестру	Вид занять/робіт	Кількість годин	Тема занять (лекції, практичного і т. н.), робіт	Література (номер за переліком), гіперпосилання
1	Аудиторні заняття			
	Лекція 1	2	Життєвий цикл створення програмного забезпечення. Метод покрокової деталізації.	[2, 4, 17]
	Лекція 2	2	Тестування та відлагодження.	[17]
	Лекція 3	2	Основи розробки інтерфейсу та створення програмної документації	[2, 4, 17]
	Лабораторна робота №1	6	Написання застосунку з графічним інтерфейсом із застосуванням методу покрокової деталізації. Тестування та відлагодження застосунку	[2, 4, 17]
	Самостійна робота			
	Опрацювання дистанційних лекцій з елементами діяльності	8	Технологія програмування та основні етапи її розвитку	[17]
		8	Основи життєвого циклу програмних засобів : Життєвий цикл розробки ПЗ	[17]
		8	Покрокова розробка програм	[17]
		8	Відлагодження програм	[17]
		12	Класифікація та короткий огляд методів тестування.	[17]
		12	Тестування комплексів програм.	[17]
		12	Процес розробки програмного забезпечення.	[17]
		12	Користувальницький інтерфейс та етапи його розробки	[17]
		10	Графічний інтерфейс	[17]
		12	Встановлення вимог. Основні методи виявлення вимог	[17]
12	Встановлення вимог. Спостереження та методи вивчення вторинної інформації	[17]		
12	Метод інженерії вимог С. Шлеєр та С. Меллора	[17]		

12	Інженерія вимог Джекобеона	[17]
10	Розробка програмної документації	[17]
8	Введення в UML, Пакети в UML.	[17]
12	Методи керування програмним проектом	[17]

Укладач А. Горбова Олександра ГОРБОВА « 01 » 02 2022 р.
 Зав. кафедри В. Горячкін Вадим ГОРЯЧКІН « 01 » 02 2022 р.
 ПІВ В. Володимир ПУЛЬПІНСЬКИЙ « 25 » 07 2022 р.

9 Методи навчання

Лекції є інформаційно-словесними з використанням електронних дидактичних демонстраційних матеріалів (презентацій). Застосовуються на лекції такі методи, як бесіда та евристична бесіда, під час яких використовується чітка система, заздалегідь визначених запитань, які сприяють активному засвоєнню студентами системи фактів, нових понять та закономірностей.

Підготовка до лекції передбачає опрацювання матеріалу попередньої лекції за конспектом, підручником [1, 2, 3, 4, 5], системою дистанційного навчання (С)ДН «Лідер» [17].

Лабораторні роботи починаються з пояснення з використанням електронних дидактичних демонстраційних матеріалів у системі СДН «Лідер». Далі виконуються обираються індивідуальні завдання та виконується лабораторна робота за індивідуальними завданням [17].

Практичні заняття починаються з пояснення з використанням електронних дидактичних демонстраційних матеріалів у системі СДН «Лідер». Далі виконуються обираються індивідуальні завдання та виконується лабораторна робота за індивідуальними завданням [17].

Опрацювання розділів програми, які не розглядаються під час лекцій, передбачає підготовку студентами конспекту відповідних тематичних питань. Для цього використовуються підручники [1, 2, 3, 4, 5], СДН «Лідер» [17], мережеві інтернет-ресурси [11-17].

Підготовка до модульного контролю передбачає опрацювання теоретичного матеріалу лекцій та виконання тестів для самоконтролю [17].

10 Методи оцінювання

Вид контролю	Метод демонстрування результатів навчання	Бал
ПК1	ЛР1. Модульне програмування. Розробка програми методом низхідного програмування.	6...10
	ЛР2. Модульне програмування. Метод висхідного програмування.	6...10
	ЛР3. Тестування та відлагодження програми	6...10
	Виконання індивідуальних завдань на практичних роботах	6...10
	Тест у СДН «Лідер» за матеріалами лекцій	6...10
ПК2	ЛР4. Розробка програми з графічним інтерфейсом	6...10
	ЛР5. Розробка простих програм засобами мови Python	6...10
	ЛР6. Використання змінюваних типів даних для реалізації рекомендаційної системи (мовою Python).	6...10
	Виконання індивідуальних завдань на практичних роботах	6...10
	Тест у СДН «Лідер» за матеріалами лекцій	6...10
	Всього	60...100

Несуть відповідальність студенти, які під час будь-якого методу оцінювання порушують принципи академічної доброчесності, тобто: **списують**, – виконують підсумкові тести із залученням зовнішніх джерел інформації, крім дозволених для використання; **обманюють** – видають звіти з лабораторних робіт, яка виконана третіми особами, як власну.

За порушення академічної доброчесності із результату, який отримав студент, вираховується 30% від максимального балу за той захід оцінювання, в якому було виявлено порушення.

Рекомендована література

1. Андон Ф.І., Коваль Г.І., Коротун Т. М., Лаврищева Є.М., Суслов В.Ю. Основи інженерії якості програмних систем. К.: Академперіодика, 2007. - 670с.
2. Бабенко Л.П., Лаврищева К.М. Основи програмної інженерії. - К.: Т-во "Знання", 2001. - 269 с.
3. Константайн Л. Локвуд Л. Розробка програмного забезпечення. - 2004.
4. Лаврищева К.М. Software Engineering комп'ютерних систем. Парадигми, технології та CASE-засоби програмування Київ. Наукова думка, 2013. – 283 с.
5. Лаврищева К.М. Програмна інженерія, Підручник, - Київ, 2008
6. Прата С. Мова програмування С++. Лекції та вправи - 6-те вид./ З. Прата – 2012.
7. Сем Канер, Джек Фолк «Тестування програмного забезпечення» – 2001
8. Шилдт Г. С++ Базовий курс / Г. Шилдт . . – 3-те вид – 2010.
9. Шилдт , Г. Повний довідник по С++ / Г. Шилдт . - 2010.
10. Шилдт , Г. С++: методики програмування Шилдт а / Г. Шилдт . - 2008.

Інформаційні ресурси

11. Базові поняття основ програмної інженерії [Електронний ресурс]. – Режим доступу: <https://uk.wikipedia.org/wiki>
12. Налагодження програм [Електронний ресурс]. – Режим доступу: <http://cppstudio.com/post/262/>
13. Тестування. [Електронний ресурс]. – Режим доступу: <https://dou.ua/forums/topic/13389/>
14. Основні положення тестування [Електронний ресурс]. – Режим доступу: <https://testitquickly.com/2010/03/09/testing-basics-by-barancev/>
15. Стадії циклу розробки ПЗ [Електронний ресурс]. – Режим доступу: <https://qalight.com.ua/baza-znaniy/stadii-tsikla-razrobotki-po/>
16. Метод інженерії вимог І. Джекобсона Електронний ресурс]. – Режим доступу: <http://posibnyky.vntu.edu.ua/bevz/35.htm>
17. Горбова О.В. Дистанційний курс Основи програмної інженерії [Електронний ресурс]. - <http://lider.diit.edu.ua/course/view.php?id=855>