

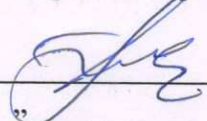
МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ  
"ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ"

Кафедра Автоматизації технологічних систем та екологічного моніторингу

«ЗАТВЕРДЖУЮ»

Голова науково-методичної комісії  
зі спеціальності 151 «Автоматизація та  
комп'ютерно-інтегровані технології»

  
П.О. Качанов  
“ \_\_\_\_\_ ” 20 \_\_\_\_ року

## РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

### Об'єктно-орієнтоване програмування

(назва навчальної дисципліни)

рівень вищої освіти \_\_\_\_\_ перший (бакалаврський) \_\_\_\_\_  
перший (бакалаврський) / другий (магістерський)

галузь знань \_\_\_\_\_ 15 "Автоматизація та приладобудування" \_\_\_\_\_  
(шифр і назва)

спеціальність \_\_\_\_\_ 151 "Автоматизація та комп'ютерно-інтегровані технології" \_\_\_\_\_  
(шифр і назва спеціальності)

освітня програма \_\_\_\_\_ 151 "Автоматизація та комп'ютерно-інтегровані технології" \_\_\_\_\_  
(шифр і назва спеціальності)

вид дисципліни \_\_\_\_\_ професійна підготовка \_\_\_\_\_  
(загальна підготовка / професійна підготовка)

форма навчання \_\_\_\_\_ денна та заочна \_\_\_\_\_  
(денна / заочна)

Харків – 2019 рік

## ЛИСТ ЗАТВЕРДЖЕННЯ

Робоча програма з навчальної дисципліни «Об'єктно-орієнтоване програмування»  
(назва дисципліни)

Розробники:

старший викладач  
(посада, науковий ступінь та вчене звання)

\_\_\_\_\_  
(підпис)

О. В. Пугановський  
(ініціали та прізвище)

Робоча програма розглянута та затверджена на засіданні кафедри

АТС та ЕМ  
(назва кафедри)

Протокол від « 26 » лютого 2019 року № 9

Завідувач кафедри АТС та ЕМ  
(назва кафедри)

\_\_\_\_\_  
(підпис)

М. О. Подустов  
(ініціали та прізвище)

## ЛИСТ ПОГОДЖЕННЯ

Назва випускової кафедри \_\_\_\_\_

Завідувач кафедри \_\_\_\_\_  
(підпис)

М.О. Подустов  
(ініціали та прізвище)

«\_\_\_\_\_» \_\_\_\_\_ 20\_\_ р.

**ЛИСТ ПЕРЕЗАТВЕРДЖЕННЯ РОБОЧОЇ НАВЧАЛЬНОЇ ПРОГРАМИ**

Дата засідання кафедри – розробника РПНД	Номер протоколу	Підпис завідувача кафедри	Підпис голови НМК (для дисциплін загальної підготовки та дисциплін професійної підготовки за спеціальністю) або завідувача випускової кафедри (для дисциплін професійної підготовки зі спеціалізації, якщо РПНД розроблена не випусковою кафедрою)

## МЕТА, КОМПЕТЕНТНОСТІ, РЕЗУЛЬТАТИ НАВЧАННЯ ТА СТРУКТУРНО-ЛОГІЧНА СХЕМА ВИВЧЕННЯ НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Мета: Навчити студентів сучасним методам та мовам об'єктно-орієнтованого програмування. Надати знання із створення програм реального часу, web-додатків і багато потокових програм.

Компетентності, відповідно до стандарту вищої освіти:

К01: Здатність застосовувати знання у практичних ситуаціях;

К04: Навички використання інформаційних та комунікаційних технологій;

К19: Здатність вільно користуватись сучасними комп'ютерними та інформаційними технологіями для вирішення професійних завдань, програмувати та використовувати прикладні та спеціалізовані комп'ютерно-інтегровані середовища для вирішення задач автоматизації;

ПРО2: Знати фізику, електротехніку, електроніку та схемотехніку, мікропроцесорну техніку на рівні, необхідному для розв'язання типових задач і проблем автоматизації;

ПРО3: Вміти застосовувати сучасні інформаційні технології та мати навички розробляти алгоритми та комп'ютерні програми з використанням мов високого рівня та технологій об'єктно-орієнтованого програмування, створювати бази даних та використовувати інтернет-ресурси.

У результаті навчання студенти повинні отримати навички самостійного створення програм для технічних систем, уміти користуватись офісним та спеціальним програмним забезпеченням, вирішувати технічні завдання вибору конфігурації обчислювальної системи та програмного забезпечення.

Структурно-логічна схема вивчення навчальної дисципліни

Попередні дисципліни:	Наступні дисципліни:
Комп'ютерні технології та програмування	Комп'ютерно-інтегровані технології
	Прикладне програмне забезпечення
Інформаційні технології і програмування	Програмне забезпечення промислових контролерів
	Комп'ютерне моделювання процесів і систем
	Основи систем автоматизованого проектування

## ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

(розподіл навчального часу за семестрами та видами навчальних занять)

Семестр	Загальний обсяг (годин) / кредитів ECTS	З них		За видами аудиторних занять (годин)			Індивідуальні завдання студентів (КП, КР, РГ, Р, РЕ)	Поточний контроль	Семестровий контроль	
		Аудиторні заняття (годин)	Самостійна робота (годин)	Лекції	Лабораторні заняття	Практичні заняття, семінари			Контрольні роботи (кількість робіт)	Залік
1	2	3	4	5	6	7	8	9	10	11
<b>4</b>	<b>180/6</b>	<b>96</b>	<b>56</b>	<b>32</b>	<b>48</b>	<b>16</b>	<b>Р</b>	<b>2</b>		<b>Е</b>

Співвідношення кількості годин аудиторних занять до загального обсягу складає 53,33 (%):

## СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

№ з/п.	Види навчальних занять (Л, ЛЗ, ПЗ, СР)	Кількість годин	Номер семестру (якщо дисципліна викладається у декількох семестрах). Назви змістових модулів. Найменування тем та питань кожного заняття. Завдання на самостійну роботу.	Рекомендована література (базова, допоміжна)
1	2	3	4	5
<b>Семестр 4</b>				
<b>Змістовий модуль № 1</b>				
<b>Основи об'єктно-орієнтованого програмування</b>				
			Тема 1 Основні принципи об'єктно-орієнтованого програмування	
1	Л	2	Лекція 1. Вступ до об'єктно-орієнтованого програмування. Сучасні мови програмування. Вимоги до розробки програмних продуктів. Платформа NET.	<b>1, 3, 4</b>
2	ЛЗ	3	Освоєння інтерфейсу середовища JetBrains PyCharm	<b>1, 4, 5</b>
3	Л	2	Лекція 2 Основи мови Python. Типи даних. Арифметичні і логічні операції	<b>1, 3, 4</b>
4	ЛЗ	3	Створення простих програмних конструкцій	<b>1, 4, 5</b>
	ПЗ	2	Області використання мови Python	<b>6, 7</b>
6	Л	2	Лекція 3 Розгалуження. Циклічні конструкції, умовні оператори	<b>1, 3, 4</b>
7	ЛЗ	3	Створення програм з розгалуженнями	<b>1, 4, 5</b>
8	Л	2	Лекція 4. Функції користувача. Області видимості змінних. Модульність програм.	<b>1, 3, 4</b>
9	ЛЗ	3	Створення програмних модулів і функцій	<b>1, 4, 5</b>
	ПЗ	2	Рекурсивні алгоритми	<b>6, 7</b>
			Тема 2 Застосування принципів ООП	<b>3, 4, 5</b>
11	Л	2	Лекція 5 Класи і об'єкти. Створення класів.	<b>1, 3, 4</b>
12	ЛЗ	3	Розробка програм з використанням класів.	<b>1, 4, 5</b>
13	Л	2	Лекція 6 Інкапсуляція	<b>1, 3, 4</b>
14	ЛЗ	3	Розробка програм на основі принципу інкапсуляції	<b>1, 4, 5</b>
	ПЗ	2	Класи і об'єкти як основа сучасних програмних продуктів	<b>6, 7</b>

16	Л	2	Лекція 7 Наслідування	<b>1, 3, 4</b>
17	ЛЗ	3	Розробка програм з використанням принципу наслідування	<b>1, 4, 5</b>
18	Л	2	Лекція 8 Поліморфізм	<b>1, 3, 4</b>
19	ЛЗ	3	Розробка програм з використанням принципу	<b>1, 4, 5</b>
	ПЗ	2	Вимоги до реалізації програмних продуктів	<b>6, 7</b>
<b>Змістовий модуль № 2 Патерни проектування</b>				
			<b>Тема 3 Використання патернів проектування</b>	<b>3, 4, 5</b>
21	Л	2	Лекція 9. Вступ до патернів проектування, Взаємодія класів і об'єктів, класифікація патернів	<b>2, 10, 11</b>
22	ЛЗ	3	Реалізація патернів у C#	<b>2, 4, 7</b>
23	Л	2	Лекція 10. Патерни, що породжують	<b>2, 10, 11</b>
24	ЛЗ	3	Створення патернів, що породжують засобами C#	<b>2, 4, 7</b>
	ПЗ	2	Визначення доцільності використання патернів	<b>2, 6, 7</b>
26	Л	2	Лекція 11 Патерни поведінки, Strategі, Observer, Command, Template	<b>2, 10, 11</b>
27	ЛЗ	3	Вирішення завдань проектування патернами поведінки, ч1.	<b>2, 4, 7</b>
28	Л	2	Лекція 12. Патерни поведінки, Iterator, State, Mediator, Memento, Visitor	<b>2, 10, 11</b>
29	ЛЗ	3	Вирішення завдань проектування патернами поведінки, ч2	<b>2, 4, 7</b>
	ПЗ	2	Визначення доцільності використання патернів поведінки	<b>2, 6, 7</b>
	Л	2	Лекція 13 Структурні патерни. Decorator, Adapter, Facade	<b>2, 10, 11</b>
32	ЛЗ	2	Вирішення завдань проектування структурними патернами, ч 1	<b>2, 4, 7</b>
33	Л	2	Лекція 14 Структурні патерни. Composite, Proxy, Bridge, Flyweight	<b>2, 10, 11</b>
34	ЛЗ	3	Вирішення завдань проектування структурними патернами, ч 2	<b>2, 4, 7</b>
	ПЗ	2	Визначення доцільності використання структурних патернів	<b>2, 6, 7</b>
36	Л	2	Лекція 15 Принципи SOLID	<b>2, 10, 11</b>



37	ЛЗ	3	Вирішення завдань проектування за принципами SOLID	<b>2, 4, 7</b>
38	Л	2	Лекція 16 Додаткові патерни	<b>2, 10, 11</b>
39	ЛЗ	3	Вибір найбільш доцільних патернів для вирішення завдань проектування програм.	<b>2, 4, 7</b>
	ПЗ	2	Стандартні і не стандартні завдання проектування програм	<b>2, 6, 7</b>
Разом за семестр		96		

## САМОСТІЙНА РОБОТА

№ з/п	Назва видів самостійної роботи	Кількість годин
1	Опрацювання лекційного матеріалу	8
2	Підготовка до практичних(лабораторних, семінарських) занять	16
3	Самостійне вивчення тем та питань, які не викладаються на лекційних заняттях	14
4	Виконання індивідуального завдання:	18
5	Інші види самостійної роботи	-
	Разом	56

## ІНДИВІДУАЛЬНІ ЗАВДАННЯ

розрахункове (створення програм розрахунку)  
(вид індивідуального завдання)

№ з/п	Назва індивідуального завдання та (або) його розділів	Терміни виконання (на якому тижні)
1	Розробка програм візуалізації математичних розрахунків.	8-16 тижні
2	Розробка програм графічної візуалізації	
3	Розробка програм роботи з масивами даних	

## МЕТОДИ НАВЧАННЯ

Теоретичний матеріал викладається на лекціях. Розгляд додаткових матеріалів відбувається на практичних заняттях. Практичні навички студенти отримують під час лабораторних занять і виконання індивідуальних завдань.

Викладання основного теоретичного матеріалу з поясненнями і наочною демонстрацією прикладів відбувається під час лекцій за допомогою презентаційного обладнання. Лекції проводяться з використанням елементів інтерактивного спілкування з аудиторією.

Лабораторні заняття проводяться з використанням необхідних технічних та програмних засобів. Групи з 2-3 студентів отримують практичне завдання. Шляхом інтерактивного спілкування з викладачем, студенти виконують завдання. Після цього відводиться час для захисту виконаної роботи, де студент пояснює прийняті рішення і показує знання теоретичної та практичної частин.

Індивідуальне завдання дає змогу закріпити отриманні знання і показати вміння студента самостійно вирішувати практичні проблеми з даної дисципліни. Результатом виконання є програма з описом її роботи, роздрукована у вигляді звіту. Захист роботи проходить шляхом розгляду і модифікації діючої програми.

## МЕТОДИ КОНТРОЛЮ

Контроль засвоєння матеріалів дисципліни складається з кількох частин. Участь студентів у обговореннях і відповіді на поставленні питання під час лекцій і практичних занять дає змогу отримати додаткові бали.

Захист лабораторних робіт показує рівень теоретичного і практичного засвоєння матеріалу і оцінюється 1-3 балами за кожну роботу.

Основним засобом контролю є модульна контрольна робота. Під час їх виконання студент показує повноту засвоєння матеріалу та вміння використовувати свої знання.

Бали отримані під час занять і на модульних контрольних підсумовуються. Якщо отримана за семестр оцінка не влаштовує студентів, вони мають змогу отримати оцінку шляхом здачі екзамену.

## РОЗПОДІЛ БАЛІВ, ЯКІ ОТРИМУЮТЬ СТУДЕНТИ, ТА ШКАЛА ОЦІНЮВАННЯ ЗНАНЬ ТА УМІНЬ (НАЦІОНАЛЬНА ТА ECTS)

Таблиця 1. Розподіл балів для оцінювання поточної успішності студента

Поточне тестування та самостійна робота			Сума
Змістовий модуль 1		Змістовий модуль 2	
T1	T2	T3	100
20	20	50	

Таблиця 2. Шкала оцінювання знань та умінь: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою
90 ... 100	A	відмінно
82 ... 89	B	добре
75 ... 81	C	
64 ... 74	D	задовільно
60 ... 63	E	
35 ... 59	FX	незадовільно з можливістю повторного складання
0 ... 34	F	незадовільно з обов'язковим повторним вивченням дисципліни

## РЕКОМЕНДОВАНА ЛІТЕРАТУРА

### Базова література

1	J. Burton Browning, Marty Alchin	Pro Python 3: Features and Tools for Professional Development	New York Business Media, New York, 2019	1
2	Svetlin Nakov & Co	Fundamentals of computer programming with C#	Sofia, 2013	1
3	Анісімов А.В. та ін.	Програмування числових методів мовою Python .	Київ : ВПЦ “Київський університет”, 2015	5
4	Рисованийий О.М.	Системне програмування	Харків : НТУ “ХПІ”, 2010	35
5	Дерев'янюк О.С.	Мова програмування Java. Лабораторний практикум з курсу “Об’єктно-орієнтоване програмування”	Харків : НТУ “ХПІ”, 2008	1

### Допоміжна література

6. Алгоритмізація та програмування : підручник / Т. В. Ковалюк ; заг. ред. В. В. Пасічник. - Львів : Магнолія 2006, 2013. - 400 с

7. Теорія програмування : навч. посібник / М. С. Бабій, О. П. Чекалов ; СумДУ. - Суми : СумДУ, 2008. - 182 с.

8. Основи програмування [Текст] : навчальний посібник / Л. І. Козак, І. В. Костюк, С. П. Стасевич. - Львів : Новий світ- 2012. - 328 с.

9. Програмування мовою C# 6.0 / І.В. Коноваленко. – Тернопіль, ТНТУ.– 2016 – 227 ст.

10. Полный справочник по C#. Г. Шилд: Пер. с англ. – М. : Издательский дом "Вильямс", 2004. – 752 с. : ил.

11. C# 4.0 на примерах. Б. Ватсон – СПб.: БХВ-Петербург, 2011. – 608 с: ил.

### ІНФОРМАЦІЙНІ РЕСУРСИ В ІНТЕРНЕТІ

1. [metanit.com](http://metanit.com) - Сайт з програмування
2. [www.mycsharp.ru](http://www.mycsharp.ru) - C# Уроки программирования с нуля. Си-шарп для чайников C#
3. [www.programmer-lib.ru/csharp.php](http://www.programmer-lib.ru/csharp.php). Обучающие уроки по C#.
4. [edurobots.ru/kurs-arduino-dlya-nachinayushhix/](http://edurobots.ru/kurs-arduino-dlya-nachinayushhix/). Учебный курс "Arduino для начинающих"
5. [arduino-diy.com](http://arduino-diy.com). Arduino-diy - проекты и обучающие уроки под Arduino