

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ОДЕСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ І. І. МЕЧНИКОВА

Кафедра механіки, автоматизації та інформаційних технологій



ЗАТВЕРДЖУЮ

Проректор з науково-педагогічної роботи

(_____)

_____ 2022 р.

РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ОП16 «Технології створення програмних засобів»

(назва навчальної дисципліни)

Рівень вищої освіти: Перший (бакалаврський)

Галузь знань: 12 «Інформаційні технології»

Спеціальність: 122 «Комп'ютерні науки»

Спеціалізація: _____

Освітньо-професійна/наукова програма: «ОСВІТНЬО-ПРОФЕСІЙНА ПРОГРАМА»

Робоча програма навчальної дисципліни «Технології створення програмних засобів» – Одеса: ОНУ, 2022. – 10 с.

Розробники:

Рачинська Алла Леонідівна, канд. фіз.-мат. наук, доцент,
Недева Ольга Анатоліївна, викладач.

Робоча програма затверджена на засіданні кафедри

механіки, автоматизації та інформаційних технологій

Протокол № 1 від “26” 08 2022 року

Завідувач кафедри _____ (підпис) (Алла РАЧИНСЬКА)

Погоджено із гарантом ОПП/ОНП

«Освітньо-професійна програма»

_____ (підпис) (Серій Шугайко)

Схвалено навчально-методичною комісією (НМК)

з інформаційних технологій

Протокол № 2 від “31” 08 2022 року

Голова НМК _____ (підпис) (Алла РАЧИНСЬКА)

Переглянуто та затверджено на засіданні кафедри

механіки, автоматизації та інформаційних технологій

Протокол № 1 від “28” 08 2023 року

Завідувач кафедри _____ (підпис) (Рачинська А.А.)

Переглянуто та затверджено на засіданні кафедри

механіки, автоматизації та інформаційних технологій

Протокол № _____ від “ _____ ” _____ 202_ року

Завідувач кафедри _____ (підпис) (_____)

1. Опис навчальної дисципліни

Найменування показників	Галузь знань, спеціальність, спеціалізація, рівень вищої освіти	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
<p>Загальна кількість:</p> <p>кредитів - 4</p> <p>годин - 120</p> <p>змістових модулів - 2</p>	<p>Галузь знань <i>12 Інформаційні технології</i> (шифр і назва)</p> <p>Спеціальність <i>122 Комп'ютерні науки</i> (код і назва)</p> <p>Спеціалізації: _____ (назва)</p> <p>Рівень вищої освіти: <i>Перший (бакалаврський)</i></p>	Обов'язкова	
		Рік підготовки:	
		2-й	
		Семестр	
		3-й	
		Лекції	
		34 год.	
		Практичні, семінарські	
		год.	
		Лабораторні	
		34 год.	
		Самостійна робота	
		52 год.	
Форма підсумкового контролю: залік			

Метою викладання дисципліни «*Технології створення програмних засобів*» є формування системи теоретичних знань і набуття практичних умінь і навичок з питань теоретичних основ та використання сучасних методології та технологій розроблення, розуміння процесу побудови програмного забезпечення із використанням патернів проектування, оволодіння методологічними та методичними основами побудови архітектури програмних застосунків в галузі програмного забезпечення.

Для досягнення мети поставлені такі основні **завдання**: вивчення понять, принципів, методології та технологій створення програмних продуктів як сукупності процесів розроблення програмних систем на засадах життєвого циклу (ЖЦ) програмного забезпечення інформаційних систем; вивчення призначення засобів об'єктного підходу до проектування програмного забезпечення інформаційних систем; забезпечити чітке володіння сучасними способами роботи з патернами проектування: породжувальними, структурними та патернами поведінки; створення програм із якісним програмним кодом при використанні технології об'єктно-орієнтованого програмування.

Процес вивчення дисципліни спрямований на формування елементів наступних компетентностей:

а) Спеціальні (фахові, предметні) компетентності:

СК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

СК9. Здатність реалізувати багаторівневу обчислювальну модель на основі архітектури клієнт-сервер, включаючи бази даних, знань і сховища даних, виконувати розподілену обробку великих наборів даних на кластерах стандартних серверів для забезпечення обчислювальних потреб користувачів, у тому числі на хмарних сервісах.

СК10. Здатність застосовувати методології, технології та інструментальні засоби для управління процесами життєвого циклу інформаційних і програмних систем, продуктів і сервісів інформаційних технологій відповідно до вимог замовника.

Програмні результати навчання:

ПР1. Застосовувати знання основних форм і законів абстрактно-логічного мислення, основ методології наукового пізнання, форм і методів вилучення, аналізу, обробки та синтезу інформації в предметній області комп'ютерних наук.

ПР10. Використовувати інструментальні засоби розробки клієнт-серверних застосувань, проектувати концептуальні, логічні та фізичні моделі баз даних, розробляти та оптимізувати запити до них, створювати розподілені бази даних, сховища та вітрини даних, бази знань, у тому числі на хмарних сервісах, із застосуванням мов веб-програмування.

ПР11. Володіти навичками управління життєвим циклом програмного забезпечення, продуктів і сервісів інформаційних технологій відповідно до вимог

і обмежень замовника, вміти розробляти проектну документацію (техніко-економічне обґрунтування, технічне завдання, бізнес-план, угоду, договір, контракт).

ПР14. Володіти навичками представлення здобутків української нації та держави, а також власних професійних (технічних, алгоритмічних, програмних) рішень (рішень команди розробників) під час супроводження продуктів галузі на етапах життєвого циклу в спілкуванні з колегами різних наукових та професійних шкіл.

У результаті вивчення навчальної дисципліни студент повинен

знати:

етапи життєвого циклу програмного продукту; як аналізувати, створювати та обирати склад та типи вимог до програмного продукту; сучасні процеси проектування та розроблення програмних продуктів; основні типи шаблонів проектування (що породжують, структурні, поведінкові); найбільш поширені шаблони, їх переваги та недоліки; критерії застосування шаблонів у конкретних ситуаціях;

вміти: аналізувати предметну область на основі об'єктно-орієнтованої методології проектування; застосовувати основні методи та інструменти розроблення програмних продуктів; будувати діаграми у мові UML для формалізації опису предметної області, для якої розроблюється програмний продукт; застосовувати шаблони, що породжують; застосовувати структурні шаблони; застосовувати поведінкові шаблони; обґрунтувати доцільність застосування того чи іншого шаблону для цієї ситуації;

3. Зміст навчальної дисципліни

Змістовий модуль 1. Технологія Windows Form Application та об'єктно-орієнтований підхід на мові C#

Тема 1. Технологія програмування та основні етапи її розвитку.

Перший етап – «стихийне» програмування. Другий етап – структурний підхід до програмування. Третій етап - об'єктний підхід до програмування. Четвертий етап - компонентний підхід та CASE-технології. Мова програмування C#: історія, специфіка, місце на ринку. Платформа розробки: .NET Framework. Модель виконання коду серед CLR. Бібліотека FCL.

Тема 2. Створення бібліотек.

Збірки. Бібліотека розв'язання квадратних рівнянь. Тестування роботи створеної бібліотеки. Вікно Оглядач об'єктів.

Тема 3. Делегати та події.

Делегат. Інкапсулювання методу. Багатоадресна передача. Події. Рекомендації щодо обробки подій у середовищі .NET Framework. Використання вбудованого делегата EventHandler.

Тема 4. Побудова графіків.

Бібліотека ZedGraph.dll. Компонент ZedGraphControl. Рефлексія.

Тема 5. Інтерфейси.

Реалізація інтерфейсів. Використання інтерфейсних посилань. Інтерфейсні властивості. Інтерфейсні індексатори. Спадкування інтерфейсів. Приховування імен за допомогою успадкування інтерфейсів. Явна реалізація членів інтерфейсу. Закрита реалізація. Як уникнути невизначеності за допомогою явної реалізації.

Змістовий модуль 2. Шаблони проектування

Тема 1. Model-View-Controller і супутні патерни в .Net.

«Оригінальний» MVC. Model-View-Presenter. Чому інтерфейс? Паттерн MVVM. Особливості реалізації MVP для Windows Forms.

Тема 2. Шаблони, що породжують.

Абстрактна фабрика (Abstract Factory). Будівельник - Builder. Фабричний метод – Factory method. Одинак – Singleton. Прототип – Prototype.

Тема 3. Структурні паттерни

Паттерн Міст (Bridge). Компонувальник (Composite). Декоратор (Decorator). Фасад (Facade).

Тема 4. Поведінкові шаблони.

Ланцюжок обов'язків - Chain of Responsibility.

4. Структура навчальної дисципліни

Назви тем	Кількість годин									
	Денна форма					Заочна форма				
	Усього	у тому числі				Усього	у тому числі			
		л	п/с	лаб	ср		л	п/с	лаб	ср
1	2	3	4	5	6	7	8	9	10	11
Змістовий модуль 1. Технологія Windows Form Application та об'єктно-орієнтований підхід на мові C#										
Тема 1. Технологія програмування та основні етапи її розвитку.	9	2		2	5					
Тема 2. Створення бібліотек.	9	2		2	5					
Тема 3. Делегати та події.	13	4		4	5					
Тема 4. Побудова графіків.	13	4		4	5					
Тема 5. Інтерфейси.	13	4		4	5					
Разом за змістовим	57	16		16	25					

модулем 1									
Змістовий модуль 2. Шаблони проектування									
Тема 1. Model-View-Controller і супутні патерни в .Net.	13	4	4	5					
Тема 2. Шаблони, що породжують.	17	6	6	5					
Тема 3. Структурні патерни.	18	4	4	10					
Тема 4. Поведінкові шаблони.	15	4	4	7					
Разом за змістовим модулем 2	63	18	18	27					
Усього годин	120	34	34	52					

5. Теми семінарських занять

Не передбачено навчальним планом

6. Теми практичних занять

Не передбачено навчальним планом

7. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Обробка інформації, що зберігається у файлах	6
2	Керування формами у додатках з інтерфейсом множини документів (додатки MDI)	6
3	Використання інтерфейсів під час розробки ПЗ.	6
4	Реалізація шаблону Абстрактна фабрика.	6
5	Реалізація шаблону Прототип.	6
6	Реалізація шаблону Ланцюжок обов'язків.	4
	Разом	34

8. Самостійна робота

№ з/п	Назва теми/ види завдань	Кількість годин
1	Створення системи заліку з використанням шаблону Model-View-Presenter /[4]	52
	Разом	52

9. Методи навчання

Словесні: лекції з аналізом конкретних прикладів, консультації.

Наочні: ілюстрація матеріалу у вигляді мультимедійних презентацій.

Практичні: лабораторні роботи.

10. Форми контролю і методи оцінювання

Методи поточного\періодичного контролю: оцінювання виконання лабораторних робіт.

Підсумковий контроль: Залік. Під час підсумкового контролю студент повинен зробити індивідуальне технічне завдання.

Критерії оцінювання на поточному та підсумковому контролі:

Відмінно: додаток розроблено повністю згідно технічного завдання. На підсумковому контролі за указаний часовий термін.

Добре: додаток має неповний функціонал користувача, але вирішує основні задачі технічного завдання. На підсумковому контролі за указаний часовий термін.

Задовільно: виконане візуальне проектування додатку, частково вирішено основні задачі технічного завдання. На підсумковому контролі за указаний часовий термін.

Незадовільно: візуальне проектування застосування виконане частково, не вирішено ні однієї задачі технічного завдання. На підсумковому контролі за указаний часовий термін.

11. Питання для підсумкового контролю

Під час проведення підсумкового контролю студент повинен розробити додаток згідно з індивідуальним технічним завданням, яке визначається білетом. Результатом виконання технічного завдання є Windows Forms Application з використанням шаблонів.

12. Розподіл балів, які отримують студенти

Змістовий модуль № 1			Змістовий модуль № 2			Підсумковий контроль (залік)	Сума балів
ЛР1	ЛР2	ЛР3	ЛР4	ЛР5	ЛР6		
10	10	10	15	15	15	25	100

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	для заліку
90 – 100	A	відмінно	зараховано
85-89	B	добре	
75-84	C		
70-74	D	задовільно	

60-69	E		
35-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

13. Методичне забезпечення

1. Електронний конспект лекцій за дисципліною «Технології створення програмних засобів».
2. Лабораторні роботи за дисципліною «Технології створення програмних засобів».

14. Рекомендована література

Основна

1. Бурлаков А.А. Об'єктно-орієнтований аналіз і проектування: метод. реком. Хмельницький: ХНУ, 2017. 136 с.
2. Daniel Solis. Illustrated C# 2012. Berkeley: APress, 2012. 732 p.
3. Авраменко В.С. Технологія програмування та створення програмних продуктів : конспект лекцій. Черкаси: ЧНУ, 2017. 192 с.
4. Алексенко О. В. Технології програмування та створення програмних продуктів: конспект лекцій. Суми : Сумський державний університет, 2013. 133 с.
5. Buschmann F., Meunier R., Rohnert H., Sommerlad P., Stal M. Pattern-Oriented Software Architecture, Volume 1, A System of Patterns. Wiley, 1996. 476 с.
6. Гнатовська Г.А. Конспект лекцій з дисципліни «Технологія створення програмних продуктів»: конспект лекцій. Одеса: Одеський державний екологічний університет, 2015. 98с.
7. Галузинський Г.П., Гордієнко І.В. Сучасні технологічні засоби обробки інформації: навч. посібник. Київ: КНЕУ, 1998. 224 с.

Додаткова

1. Николайчук Я. М., Возна Н. Я., Пітух І. Р. Проектування спеціалізованих комп'ютерних систем : навч. посібник Тернопіль : ТзОВ «Терно-граф», 2010. 392 с.
2. Шаховська Н. Б., Литвин В. В. Проектування інформаційних систем : навч. посібник Львів : Магнолія-2006, 2011. 380 с.
3. Л. П. Бабенко, К. М. Лавріщева Основи програмної інженерії : навч. посіб. Київ : Знання, 2001. 270 с.
4. С. Alexander et al. A pattern language: towns, buildings, construction. New York: Oxford University Press, 1977. 120 с.

5. Standard ECMA-334. C# Language Specification. 4th Edition. Ecma International. June 2006 URL: https://www.ecma-international.org/wp-content/uploads/ECMA-334_4th_edition_june_2006.pdf
6. C# Reference, сайт розробників MSDN. URL: <https://code.visualstudio.com/docs/languages/csharp> (дата звернення 11.02.2023)