

**СИЛАБУС**  
**навчальної дисципліни**  
**«Шаблони об'єктно-орієнтованого програмування»**

<b>рівень вищої освіти</b>	перший (бакалаврський) рівень
<b>галузь знань</b>	01 Освіта/Педагогіка
<b>спеціальність (предметна спеціальність)</b>	014.04 Середня освіта (Математика)
<b>освітня програма</b>	Математика та інформатика
<b>вид дисципліни</b>	вибіркова
<b>факультет</b>	факультет математики і інформатики

**Розробники програми:**

Світлана Ігнатович, доктор фізико-математичних наук, доцент, професор кафедри прикладної математики, [s.ignatovich@karazin.ua](mailto:s.ignatovich@karazin.ua)

Олена Перепелиця, старший викладач кафедри вищої математики та інформатики

**Викладач дисципліни:** Перепелиця Олена Миколаївна, старший викладач закладу вищої освіти кафедри вищої математики та інформатики факультету математики і інформатики, [olena.perepelitsa@karazin.ua](mailto:olena.perepelitsa@karazin.ua)

**1. Опис навчальної дисципліни**

**1.1. Мета викладання навчальної дисципліни** «Шаблони в об'єктно-орієнтованому програмуванні» є формування теоретичних знань та практичних навичок з розуміння та застосування основ шаблонного проектування, ознайомлення з основами узагальненого програмування.

**1.2. Основні завдання вивчення дисципліни** «Шаблони в об'єктно-орієнтованому програмуванні» є навчити студентів оперувати теоретичними аспектами технологій шаблонного проектування; володіти практичними вміннями і навичками розробки додатків з використанням патернів на C++; створювати шаблони функцій і класів та застосовувати їх.

**1.3. Кількість кредитів:** 4.

**1.4. Загальна кількість годин:** 120 год.

**2. Тематичний план навчальної дисципліни**

*Розділ 1. Об'єктно-орієнтоване програмування. (4)*

*Тема 1. Основні терміни об'єктного програмування.*

Зміст: Об'єкти, класи. Ієрархії класів. Наріжні камені ООП – абстракція, інкапсуляція, спадкування, поліморфізм, зв'язки між об'єктами.

*Розділ 2. Принципи проектування. (19)*

*Тема 1. Вступ до патернів.*

Зміст: поняття патернів. Історія патернів. Користь патернів. Критика патернів. Класифікація патернів.

*Тема 2. Принципи проектування.*

Зміст: Якості хорошої архітектури. Повторне використання програмної архітектури та коду. Базові принципи проектування. Інкапсуляція на рівні класу, інкапсуляція на рівні методу. Програмування на рівні інтерфейсу.

### *Тема 3. Принципи SOLID.*

Зміст: Принцип єдиного обов'язку (Single Responsibility Principle). Принцип відкритості/закритості (Open/Closed Principle). Принцип підстановки Лісков (Liskov Substitution Principle). Принцип поділу інтерфейсу (Interface Segregation Principle). Принцип інверсії залежностей (Dependency Inversion Principle).

## ***Розділ 3. Каталог патернів. Породжувальні патерни проектування. (23)***

### *Тема 1. Породжувальні патерни проектування. Фабричний метод (Factory Method).*

Зміст: Проблема. Рішення. Структура. Псевдокод. Застосування. Кроки реалізації. Переваги та недоліки. Відносини з іншими патернами.

### *Тема 2. Породжувальні патерни проектування. Абстрактна фабрика (Abstract Factory). Будівельник (Builder).*

Зміст: Проблема. Рішення. Структура. Псевдокод. Застосування. Кроки реалізації. Переваги та недоліки. Відносини з іншими патернами.

### *Тема 3. Породжувальні патерни проектування. Прототип (Prototype). Одинак (Singleton).*

Зміст: Проблема. Рішення. Структура. Псевдокод. Застосування. Кроки реалізації. Переваги та недоліки. Відносини з іншими патернами.

## ***Розділ 4. Каталог патернів. Структурні патерни проектування. (28)***

### *Тема 1. Структурні патерни проектування. Адаптер (Wrapper, Обгортка, Adapter). Міст (Bridge).*

Зміст: Проблема. Рішення. Структура. Псевдокод. Застосування. Кроки реалізації. Переваги та недоліки. Відносини з іншими патернами.

### *Тема 2. Структурні патерни проектування. Компонувальник (Дерево, Composite). Декоратор (Wrapper, Обгортка, Decorator).*

Зміст: Проблема. Рішення. Структура. Псевдокод. Застосування. Кроки реалізації. Переваги та недоліки. Відносини з іншими патернами.

### *Тема 3. Структурні патерни проектування. Фасад (Facade). Легковаговик (Пристосованець, Кеш, Flyweight).*

Зміст: Проблема. Рішення. Структура. Псевдокод. Застосування. Кроки реалізації. Переваги та недоліки. Відносини з іншими патернами.

### *Тема 4. Структурні патерни проектування. Замісник (Proxy).*

Зміст: Проблема. Рішення. Структура. Псевдокод. Застосування. Кроки реалізації. Переваги та недоліки. Відносини з іншими патернами.

## ***Розділ 5. Каталог патернів. Поведінкові патерни проектування. (12)***

### *Тема 1. Поведінкові патерни проектування. Ланцюжок обов'язків (Ланцюг відповідальностей, CoR, Chain of Command, Chain of Responsibility).*

Зміст: Проблема. Рішення. Структура. Псевдокод. Застосування. Кроки реалізації. Переваги та недоліки. Відносини з іншими патернами.

*Тема 2. Поведінкові патерни проектування. Команда (Дія, Транзакція, Action, Command). Ітератор (Iterator).*

Зміст: Проблема. Рішення. Структура. Псевдокод. Застосування. Кроки реалізації. Переваги та недоліки. Відносини з іншими патернами.

*Тема 3. Поведінкові патерни проектування. Посередник (Intermediary, Controller, Mediator). Знімок (Memento).*

Зміст: Проблема. Рішення. Структура. Псевдокод. Застосування. Кроки реалізації. Переваги та недоліки. Відносини з іншими патернами.

*Тема 4. Поведінкові патерни проектування. Спостерігач (Видавець-Підписник, Слухач, Observer). Стан (State).*

Зміст: Проблема. Рішення. Структура. Псевдокод. Застосування. Кроки реалізації. Переваги та недоліки. Відносини з іншими патернами.

*Тема 5. Поведінкові патерни проектування. Стратегія (Strategy). Шаблонний метод (Template Method).*

Зміст: Проблема. Рішення. Структура. Псевдокод. Застосування. Кроки реалізації. Переваги та недоліки. Відносини з іншими патернами.

*Тема 6. Поведінкові патерни проектування. Відвідувач (Visitor).*

Зміст: Проблема. Рішення. Структура. Псевдокод. Застосування. Кроки реалізації. Переваги та недоліки. Відносини з іншими патернами.

### 3. Методи навчання

Різні групи методів: 1) методи організації та здійснення навчально-пізнавальної діяльності (розповідь-пояснення, бесіда, лекція, ілюстрація, демонстрація, вправи, індукція, дедукція), 2) методи контролю/самоконтролю, корекції/самокорекції за ефективністю навчально-пізнавальної діяльності (виконання самостійних завдань, виконання групових завдань), 4) інтегровані методи (комплексне поєднання кількох методів).

### 4. Методи контролю

Контроль за відвідуванням занять, усне опитування під час лекцій та практичних занять, перевірка домашніх завдань, курсової роботи, заключний семестровий контроль у формі екзамену.

### 5. Схема нарахування балів

		Поточний контроль, самостійна робота, індивідуальні завдання															Індивідуальне завдання	Разом	Екзамен	Сума
Розділ 1	Розділ 2			Розділ 3			Розділ 4				Розділ 5									
T1	T1	T2	T3	T1	T2	T3	T1	T2	T3	T4	T1	T2	T3	T4	T5	T6				
1	1	3	3	2	3	3	3	3	2	3	3	3	3	3	3	3	15	60	40	100

T1, T2 ... – теми розділів.

Для допуску до складання підсумкового контролю (екзамен) здобувач вищої освіти повинен набрати не менше 10 балів з навчальної дисципліни під час поточного контролю, самостійної роботи, індивідуального завдання.

### Шкала оцінювання

Сума балів за всі види навчальної діяльності протягом семестру	Оцінка	
	для чотирирівневої шкали оцінювання	для дворівневої шкали оцінювання
90 – 100	відмінно	зараховано
70-89	добре	
50-69	задовільно	
1-49	незадовільно	не зараховано

## 6. Рекомендована література

### Основна література

1. Олександр Швець. Занурення в патерни проектування. © Олександр Швець, Refactoring.Guru – 2022.
2. Еріх Гамма, Річард Хелм, Ральф Джонсон, Джон Вліссідес. Design Patterns: Elements of Reusable Object-Oriented Software – 1995.
3. Robert C. Martin: Agile Software Development, Principles, Patterns, and Practices. 2021
4. Bjarne Stroustrup (The Creator of C++) The C++ Programming Language C++11, 4th Edition. 2016

### Допоміжна література

1. Глинський Я. М., Анохін В. Є., Ряжська В. А. C++ і C++ Builder. Навчальний посібник. Львів – 2006.

### Посилання на інформаційні ресурси в Інтернеті, відео-лекції, інше методичне забезпечення

1. <https://codeguida.com/>
2. <https://refactoring.guru/>
3. <https://www.bestprog.net/uk/>
4. <https://github.com/septimomend/DesignPatterns-empl#articles>