



# ПРОГРАМУВАННЯ. КУРСОВА РОБОТА

## Робоча програма навчальної дисципліни (Силабус)

### Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>121 Інженерія програмного забезпечення</i>
Освітня програма	<i>Інженерія програмного забезпечення мультимедійних та інформаційно-пошукових систем</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>Очна (денна)</i>
Рік підготовки, семестр	<i>2 рік навчання, 4 семестр</i>
Обсяг дисципліни	<i>Самостійна робота: 30 год.</i>
Семестровий контроль/ контрольні заходи	<i>Залік</i>
Розклад занять	<i>Не передбачено</i>
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	<i>Комісія з прийому захистів КР: к.т.н., доцент, Заболотня Тетяна Миколаївна, <a href="mailto:tetiana.zabolotnia@gmail.com">tetiana.zabolotnia@gmail.com</a> асистент Юсин Яків Олексійович <a href="mailto:yusin.yakiv@gmail.com">yusin.yakiv@gmail.com</a></i>
Розміщення курсу	<i>Google disk</i>

### Програма навчальної дисципліни

#### 1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

**Метою** виконання курсової роботи з дисципліни «Програмування. Курсова робота» є закріплення, поглиблення й узагальнення теоретичних знань щодо методів і технологій об'єктно-орієнтованого програмування, отриманих студентами за час навчання, а також їх застосування до комплексного вирішення конкретного фахового завдання, зокрема набуття навичок побудови програмних продуктів (компонентів) багаторазового використання на основі шаблонів проєктування.

**Завданнями** даної курсової роботи є:

- сприяння набуттю студентами навичок щодо самостійної роботи над навчально-методичною, технічною, монографічною та періодичною літературою й іншими інформаційними джерелами;
- відпрацювання вмінь аналізувати та алгоритмізувати поставлену задачу, оцінювати доцільність застосування шаблонів проєктування для створення програмного забезпечення та обґрунтовувати обрану структурну організацію проєкту;
- набуття досвіду щодо розроблення програмних класів (компонентів), придатних для повторного використання відповідно до об'єктно-орієнтованого підходу;
- розвиток навичок програмування, у т.ч. створення програмного забезпечення зі складним інтерфейсом користувача, та роботи у середовищах програмування;

– вироблення та закріплення вміння документування новоствореного програмного коду, написання технічної документації до розробленого проєкту (зокрема, технічного завдання, описів модулів та класів, UML- діаграм шаблонів, інструкції користувача).

Оскільки курсова робота є одним з видів індивідуальних завдань, до її завдань також входить формування у студентів навичок самостійної роботи взагалі в навчальній, професійній діяльності, здатності брати на себе відповідальність, самостійно вирішувати проблеми, знаходити конструктивні рішення, вихід з кризових ситуацій тощо. Крім того, виконання курсової роботи сприяє набуттю досвіду творчої та дослідницької діяльності з вирішення поставлених завдань, вміння виробляти власне ставлення до розв'язання практичної проблеми.

**Предметом** дисципліни «Програмування. Курсова робота» є процес розроблення об'єктно-орієнтованого програмного забезпечення з використанням шаблонів проєктування.

В контексті освітньої програми виконання курсової роботи з дисципліни «Програмування. Курсова робота» формує у здобувачів освіти **фахові компетентності (ФК)** та **програмні результати навчання (ПРН)**, необхідних для розв'язання практичних задач професійної діяльності, пов'язаної із проєктуванням та розробленням об'єктно-орієнтованого програмного забезпечення:

- **ФК01** Здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення;
- **ФК02** Здатність брати участь у проєктуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування;
- **ФК03** Здатність розробляти архітектури, модулі та компоненти програмних систем;
- **ФК05** Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу;
- **ФК07** Володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних;
- **ФК08** Здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення;
- **ФК13** Здатність обґрунтовано обирати та освоювати інструментарій з розроблення та супроводження програмного забезпечення;
- **ФК14** Здатність до алгоритмічного та логічного мислення;
- **ФК17** Здатність розробляти програмне забезпечення інформаційно-пошукових систем;
- **ФК19** Здатність розробляти програмне забезпечення мультимедійних та мультимедійних систем;
  
- **ПРН01** Аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки;
- **ПРН03** Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення;
- **ПРН04** Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення;
- **ПРН06** Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення;
- **ПРН07** Знати і застосовувати на практиці фундаментальні концепції, парадигми і основні принципи функціонування мовних, інструментальних і обчислювальних засобів інженерії програмного забезпечення;
- **ПРН08** Знати та вміти розробляти людино-машинний інтерфейс;
- **ПРН10** Проводити передпроєктне обстеження предметної області, системний аналіз об'єкта проєктування;

- **ПРН12** Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення;
- **ПРН13** Знати і застосовувати методи розроблення алгоритмів, конструювання програмного забезпечення та структур даних і знань;
- **ПРН14** Застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення;
- **ПРН15** Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення;
- **ПРН16** Мати навички програмного розроблення, погодження оформлення і випуску всіх видів програмної документації.
- **ПРН23** Вміти документувати та презентувати результати розроблення програмного забезпечення.
- **ПРН38** Вміти застосовувати технології програмування для розроблення програмного забезпечення мультимедійних та інформаційно-пошукових систем.
- **ПРН42** Знати основні моделі подання текстової та мультимедійної інформації та способи її попереднього оброблення для застосовування при проектуванні інформаційно-пошукових систем.
- **ПРН43** Знати та вміти використовувати на практиці існуючі програмні ресурси та бібліотеки для оброблення текстової інформації та мультимедійних даних в інформаційно-пошукових системах.
- **ПРН44** Володіти найбільш поширеними мовами запитів, що використовуються при розробленні інформаційно-пошукових систем.

## **2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)**

Успішному виконанню курсової роботи з дисципліни «Програмування. Курсова робота» передуює вивчення дисциплін «Алгоритми та структури даних», «Основи програмування» та «Програмування 1. Об'єктно-орієнтоване програмування та шаблони проектування» навчального плану підготовки бакалаврів за спеціальністю 121 Інженерія програмного забезпечення.

Отримані теоретичні знання та практичні уміння є необхідними для подальшого успішного виконання студентами курсових робіт, проходження практики, а також виконання дипломного проекту в рамках навчального плану підготовки бакалаврів за спеціальністю 121 Інженерія програмного забезпечення.

## **3. Типове завдання на курсову роботу**

Розробити програмне забезпечення згідно з обраною темою та оформити документацію на виконану розробку відповідно до вимог, наведених нижче.

Розроблюване програмне забезпечення має базуватися на реалізації шаблонів проектування:

- мінімальна кількість застосованих шаблонів – 7 шт;
- шаблони можуть бути як незалежними один від одного, так і поєднаними у більш складні структури (по 2 шаблони та більше);
- складність обраних для реалізації шаблонів впливатиме на оцінювання роботи: використання найпростіших шаблонів (наприклад, «Одинак») без об'єктивної необхідності призведе до зниження оцінки за роботу.

*Програмне забезпечення повинне мати інтерфейс користувача (консольний чи графічний), за допомогою якого можна керувати роботою програми: варіант функціонування програми, коли виконується один сценарій, після чого програма завершує свою роботу, є неприпустимим.*

*Мова програмування має бути заздалегідь узгоджена з керівником курсової роботи та затверджена у технічному завданні.*

*Приклади тематик для курсової роботи:*

1. *Фруктовий сад*
2. *Шахи*
3. *Морський бій*
4. *Текстовий редактор*
5. *Графічний редактор*
6. *Музичний редактор*
7. *Тролейбусне депо*
8. *Робота з банківськими рахунками*
9. *Бібліотека*
10. *Лікарня*
11. *Деканат*
12. *Продаж квитків на залізничному вокзалі*
13. *Облік пасажирів аеропорту*
14. *Облік даних щодо інфраструктури району міста*
15. *ЖЕК*
16. *Олімпіада*
17. *Бухгалтерія підприємства*
18. *Облік товару на складі*
19. *Облік даних щодо меню ресторану*
20. *Ремонтна майстерня широкого профілю*
21. *Облік робочого часу працівників*
22. *Записник – дати, телефони, зустрічі, завдання*
23. *Облік матеріальних цінностей*
24. *Файловий менеджер*
25. *Гра «П'ятнашки»*

#### **4. Організація виконання та захисту курсової роботи**

*Основними етапами виконання курсової роботи з дисципліни «Програмування. Курсова робота» є:*

- вибір теми;*
- підбір та вивчення літератури;*
- оформлення технічного завдання;*
- проведення агрегування класів об'єктів з обраної тематики, визначення структури розроблюваного програмного забезпечення, міжмодульних зв'язків, вибір шаблонів проектування для реалізації в межах курсової роботи;*
- підготовка першого варіанту структурної схеми програми, діаграм класів та перше узгодження з керівником;*
- доопрацювання структури модулів та класів з урахуванням пропозицій керівника;*
- розроблення алгоритмів роботи та інтерфейсу відповідного програмного забезпечення;*
- друге узгодження результатів роботи з керівником;*
- виконання програмної реалізації;*
- демонстрування першого варіанту розробленого програмного забезпечення, третє узгодження з керівником;*
- доопрацювання роботи відповідно до зауважень керівника, тестування програми;*

- написання та оформлення тексту пояснювальної записки до курсової роботи;
- захист курсової роботи.

Курсова робота виконується самостійно, але не втрачаючи при цьому зв'язку з керівником. Керівник погоджує технічне завдання на курсову роботу та графік її виконання, здійснює контроль за ходом роботи, надає студенту необхідну консультативну допомогу у проведенні досліджень та розробленні програмного забезпечення, дає висновок про допуск роботи до захисту.

Дозволяється спільна робота студентів над однією темою, але в такому випадку у кожного студента повинен бути реалізований свій набір шаблонів.

До захисту курсової роботи допускаються студенти, які виконали її в повному обсязі та у встановлені терміни. У разі недопущення курсової роботи до захисту студент повинен врахувати зауваження керівника курсової роботи та доопрацювати її з повторною подачею керівникові.

Захист курсової роботи проходить за встановленим графіком захисту. Для захисту студент повинен пред'явити:

- оформлену пояснювальну записку;
- розроблене програмне забезпечення та вихідний код (обов'язковим є демонстрування роботи програми на комп'ютері/ноутбуку студента).

Для захисту курсової роботи кожному студенту надається 7-10 хвилин. У своїй доповіді автор роботи має викласти основні положення роботи, обґрунтувати та продемонструвати структурно-алгоритмічну організацію розробленого програмного забезпечення, провести демонстрацію роботи програми, що засвідчить факт виконання студентом поставлених завдань.

Під час захисту роботи, написаної на комплексну тему, у доповіді обов'язково потрібно окреслити особистий внесок студента у розроблення тих чи інших питань.

Після доповіді студенту можуть бути поставлені запитання з тематики його роботи, на які він повинен дати чітку й обґрунтовану відповідь. Студент має бути готовим до дискусії, володіти понятійним апаратом, знати основний зміст праць, що наводяться в бібліографічному покажчику.

За результатами захисту курсової роботи, яка відповідає викладеним у даних методичних рекомендаціях вимогам, студент отримує оцінку, яку викладач виставляє у відомість.

## **5. Графік виконання курсового проєкту**

1 Затвердження теми курсової роботи. Опрацювання відповідної літератури. Розроблення та узгодження технічного завдання, оформлення аркуша завдання – до 1 березня.

2 Аналіз постановки задачі – до 9 березня.

3 Вибір та дослідження методів створення програми, вибір відповідних структур даних та шаблонів проєктування. Перше узгодження з керівником – до 15 березня.

4 Доопрацювання структури модулів та класів з урахуванням пропозицій керівника. Проєктування інтерфейсу користувача – до 31 березня.

5 Розроблення основних алгоритмів роботи програми та проєктування інтерфейсу. Друге узгодження з керівником – до 15 квітня.

6 Програмна реалізація – до 10 травня.

7 Демонстрування першого варіанту. Третє узгодження з керівником – до 20 травня.

8 Тестування програми. Аналіз результатів. Підготовка матеріалів КР та оформлення документації – до 31 травня.

9 Захист курсової роботи – до 15 червня.

## 6. Навчальні матеріали та ресурси

### Базова література:

1. Бублик В.В. Об'єктно-орієнтоване програмування. – К.: ІТ-книга, 2015. – 624 с.:іл.
2. Шаблони проектування: методичні вказівки до виконання лабораторних робіт з дисципліни «Об'єктно-орієнтоване програмування» [Електронне видання] / Т.М.Заболотня. К.: НТУУ «КПІ», 2015. - 154с. <https://ela.kpi.ua/handle/123456789/11657>.
3. Будай А. Дизайн-патерни – просто, як двері [Електронне видання] / А.Будай. URL: <http://designpatterns.andriybuday.com/>.
4. Head First. Патерни проектування / Ерік Фрімен, Елізабет Робсон, Кеті Сьерра і Берт Бейтс; пер. з англ. Г. Якубовська – Харків : ВД «Фабула», 2020. – 672 с.
5. Чистий код: створення і рефакторинг за допомогою Agile / Мартін Роберт; пер. з англ. І. Бондар-Терещенко. – Харків : ВД «Ранок» : Фабула, 2019. – 448 с.
6. Чому SOLID — важлива складова мислення програміста [Електронний ресурс] / І.Бранець URL: <https://dou.ua/lenta/articles/solid-principles/>
7. Рефакторинг.Гуру [Електронний ресурс] / О.Швець URL: <https://refactoring.guru/>

### Допоміжна література:

8. B.Perkins, J.V.Hammer, J.D.Reid Beginning C# 7 Programming with Visual Studio 2017 ISBN: 978-1-119-45868-5 John Wiley & Sons, Inc, 2018 – 912pp.
9. M.Olsson C# 7 Quick Syntax Reference: A Pocket Guide to the Language, APIs, and Library ISBN-13 (pbk): 978-1-4842-3816-5, 2018 – 180pp.
10. M.Fowler Refactoring: Improving the Design of Existing Code (2nd Edition) (Addison-Wesley Signature Series (Fowler)) 2nd Edition ISBN-13: 978-0134757599 – 2018. – 448 pp.
11. Bishop C# 3.0 Design Patterns ISBN 10: 0-596-52773-X, ISBN 13: 978-0-596-52773-0, O'Reilly, 2008. – 290pp.
12. Tour the Visual Studio IDE 15 [Електронний ресурс] / URL: <https://learn.microsoft.com/en-us/visualstudio/ide/quickstart-ide-orientation?view=vs-2022>

Матеріали знаходяться у вільному доступі в Інтернеті.

## Політика та контроль

### 7. Політика навчальної дисципліни (освітнього компонента)

- Дотримання політики академічної доброчесності.
- При захисті курсові роботи повинні бути зроблені згідно до затвердженої теми здобувача освіти.
- Штрафні бали нараховуються за: плагіат. У випадку виявлення факту несамостійного виконання роботи (код програми не відповідає варіанту завдання, ідентичність коду програми серед різних робіт) студент отримує мінус 30 балів.

### 8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Семестровий рейтинг з курсової роботи складається з рейтингових балів (див. табл. 1) і не перевищує 100 балів.

*Система нарахування рейтингових балів з курсової роботи*

<i>Критерій оцінювання</i>	<i>Бали</i>
<i>Теоретична складова</i>	
<i>1. Складність розробки обраної теми та ступінь творчого характеру роботи</i>	<i>4</i>
<i>2. Повнота опрацювання джерел з об'єктно-орієнтованого програмування, зокрема, з шаблонів проектування</i>	<i>2</i>
<i>3. Повнота об'єктної моделі, побудованої за тематикою дослідження</i>	<i>5</i>
<i>4. Складність обраних для реалізації шаблонів проектування</i>	<i>3</i>
<i>5. Якість викладу теоретичних положень щодо шаблонів проектування, ступінь обґрунтованості вибору шаблонів для реалізації програмного забезпечення за тематикою КР</i>	<i>8</i>
<i>6. Якість аргументації та ґрунтовність опису реалізованої структури ПЗ, алгоритмів та методів</i>	<i>8</i>
<i>Разом за теоретичну складову:</i>	<i>30</i>
<i>Програмна складова</i>	
<i>1. Ступінь відповідності реалізації програмного коду основним принципам ООП</i>	<i>8</i>
<i>2. Якість програмної реалізації шаблонів проектування</i>	<i>8</i>
<i>3. Ступінь відповідності функціональності розробленого програмного забезпечення вимогам, сформульованим у технічному завданні</i>	<i>5</i>
<i>4. Якість створення візуального контролю за вхідними, проміжними та результуючими даними</i>	<i>3</i>
<i>5. Якість коментарів у програмі</i>	<i>3</i>
<i>6. Наявність достатньої кількості тестів ПЗ</i>	<i>3</i>
<i>Разом за програмну складову:</i>	<i>30</i>
<i>Оформлення документації до курсової роботи</i>	
<i>1. Загальний вигляд документації</i>	<i>1</i>
<i>2. Правильність оформлення титульного аркуша</i>	<i>1</i>
<i>3. Правильність оформлення технічного завдання</i>	<i>2</i>
<i>4. Ступінь лаконічного і чіткого викладу технічного матеріалу, у тому числі описів шаблонів, модулів, методів, алгоритмів</i>	<i>4</i>
<i>5. Якість опису інтерфейсу користувача</i>	<i>3</i>
<i>6. Якість оформлення прикладів результатів роботи програми</i>	<i>2</i>
<i>7. Загальна відповідність пояснювальної записки вимогам до оформлення роботи</i>	<i>2</i>
<i>Разом за оформлення документації:</i>	<i>15</i>
<i>Захист роботи</i>	
<i>1. Ступінь володіння матеріалом з теоретичних і практичних питань КР</i>	<i>15</i>
<i>2. Глибина і правильність відповідей на запитання</i>	<i>10</i>
<i>Разом за захист роботи:</i>	<i>25</i>
<i>РАЗОМ:</i>	<i>100</i>

Рейтинговий бал студента з курсової роботи визначається остаточно за результатами захисту в термін, що передбачається графіком, виходячи з максимальної суми – 100 балів. Студент вважається таким, що захистив курсову роботу, якщо він набрав 60 і більше балів.

Заохочувальні бали можуть бути нараховані за самостійне виконання курсової роботи за завданням підвищеної складності (до 7 балів), а також за створення довідникового модуля до розробленого в межах курсової роботи програмного забезпечення (до 3 балів).

Студент допускається до захисту, якщо його рейтинг з урахуванням штрафних та заохочувальних балів складає не менше, ніж 45 балів.

З урахуванням штрафних і заохочувальних балів сумарний рейтинг не може перевищувати 100 балів.

Семестровий контроль: залік.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

<i>Кількість балів</i>	<i>Оцінка</i>
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

**Робочу програму навчальної дисципліни (силабус):**

Складено к.т.н., доц., Заболотня Т.М.

Ухвалено кафедрою ПЗКС (протокол № 12 від 26.04.2023 р.)

Погоджено Методичною комісією факультету прикладної математики (протокол № 10 від 26.05.2023 р.)