

**НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ БІОРЕСУРСІВ
І ПРИРОДОКОРИСТУВАННЯ УКРАЇНИ**
Кафедра комп'ютерних наук

«ЗАТВЕРДЖЕНО»
Факультет інформаційних технологій
“12” червня 2025 р.

**РОБОЧА ПРОГРАМА
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**
Конструювання програмного забезпечення

Галузь знань Інформаційні технології

Спеціальність Інженерія програмного забезпечення

Освітня програма Інженерія програмного забезпечення

Факультет інформаційних технологій

Розробники: __ст.викладач Бородкін Г.О.

(посада, науковий ступінь, вчене звання)

Опис навчальної дисципліни "Конструювання програмного забезпечення" —

(до 1000 друкованих знаків)

У даній дисципліні висвітлюються питання, які виникають під час конструювання програмного забезпечення та процесів, пов'язаних із ним. У курсі розглянуто базові підходи до конструювання, різні способи керування ним та практичні аспекти, що виникають у процесі конструювання програмного забезпечення. Також увага приділяється проектуванню за допомогою «шаблонів проектування» та UML, як засобу їх візуального подання. Завдання дисципліни "Конструювання програмного забезпечення" - надати студентам знання в сфері реалізації задач автоматизації обробки інформації, автоматизації керування об'єктами, в тому числі, сільськогосподарськими, за допомогою комп'ютерної техніки. Такі знання майбутній спеціаліст зможе застосовувати як при подальшому навчанні, так і після отримання вищої освіти у своїй професійній діяльності.

По завершенню курсу студенти захищають курсовий проєкт.

| Галузь знань, спеціальність, освітня програма, освітній ступінь | | |
|---|--|--------|
| Освітній ступінь | бакалавр | |
| Спеціальність | 121 Інженерія програмного забезпечення | |
| Освітня програма | Інженерія програмного забезпечення | |
| Характеристика навчальної дисципліни | | |
| Вид | обов'язкова | |
| Загальна кількість годин | 150 | |
| Кількість кредитів ECTS | 5 | |
| Кількість змістових модулів | 2 | |
| Курсовий проєкт (робота) (за наявності) | + | |
| Форма контролю | екзамен | |
| Показники навчальної дисципліни для денної та заочної форм здобуття вищої освіти | | |
| | Форма здобуття вищої освіти | |
| | денна | заочна |
| Курс (рік підготовки) | 3 | |
| Семестр | 5 | |
| Лекційні заняття | 30 год. | год. |
| Практичні, семінарські заняття | год. | год. |
| Лабораторні заняття | 30 год. | год. |
| Самостійна робота | 90 год. | год. |
| Кількість тижневих аудиторних годин для денної форми здобуття вищої освіти | 4 год. | |

1. Мета, компетентності та програмні результати навчальної дисципліни

Мета вивчення курсу «Конструювання програмного забезпечення» - формування бази теоретичних знань та умінь щодо сучасних методів та засобів конструювання програмних систем.

Набуття компетентностей:

інтегральна компетентність (ІК):

Здатність розв'язувати складні задачі і проблеми інженерії програмного забезпечення, що передбачає проведення досліджень з елементами наукової новизни та/або здійснення інновацій в умовах невизначеності вимог.

загальні компетентності (ЗК):

K05. Здатність вчитися і оволодівати сучасними знаннями.

спеціальні (фахові) компетентності (СК):

K14. Здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування.

K17. Здатність дотримуватися специфікацій, стандартів, правил і рекомендацій в професійній галузі при реалізації процесів життєвого циклу.

K23. Здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення.

K25. Здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення.

Програмні результати навчання (ПРН):

ПР03. Знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення.

ПР04. Знати і застосовувати професійні стандарти і інші нормативно-правові документи в галузі інженерії програмного забезпечення.

ПР06. Уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення, враховуючи специфіку природоохоронної галузі.

ПР08. Вміти розробляти людино-машинний інтерфейс.

ПР09. Знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення.

ПР12. Застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення.

ПР17. Вміти застосовувати методи компонентної розробки програмного забезпечення.

2. Програма та структура навчальної дисципліни

| Назви змістових модулів і тем | Кількість годин | | | | | | | | | | | | | |
|--|-----------------|------------|--------------|---|-----|-----|------|--------------|--------------|---|-----|-----|------|--|
| | денна форма | | | | | | | заочна форма | | | | | | |
| | тижні | усьо го | у тому числі | | | | | усього | у тому числі | | | | | |
| | | | л | п | лаб | інд | с.р. | | л | п | лаб | інд | с.р. | |
| Модуль 1. Загальні відомості про конструювання програмного забезпечення | | | | | | | | | | | | | | |
| Тема 1. Вступ до дисципліни «Конструювання програмного забезпечення» | 1 | 8 | 2 | | | | 6 | | | | | | | |
| Тема 2. Введення в конструювання програмного забезпечення. | 1 | 12 | 2 | | 4 | | 6 | | | | | | | |
| Тема 3. Загальні підходи до проектування програмних систем | 1 | 12 | 2 | | 4 | | 6 | | | | | | | |
| Тема 4. Техніка моделювання архітектури ПЗ. | 1 | 8 | 2 | | | | 6 | | | | | | | |
| Тема 5. Концепції сервіс-орієнтованої архітектури ПЗ. | 1 | 12 | 2 | | 4 | | 6 | | | | | | | |

| | | | | | | | | | | | | | |
|--|------------|----|-----------|--|-----------|-----------|--|--|--|--|--|--|--|
| Тема 6. RESTful та SOAP Web-сервіси. | 1 | 8 | 2 | | | 6 | | | | | | | |
| Тема 7. Документування архітектури ПЗ. | 1 | 12 | 2 | | 4 | 6 | | | | | | | |
| Разом за модулем 1 | 72 | | 14 | | 16 | 42 | | | | | | | |
| Модуль 2. Керування конструюванням програмного забезпечення | | | | | | | | | | | | | |
| Тема 8. Методи об'єктного аналізу і моделювання. | 1 | 8 | 2 | | | 6 | | | | | | | |
| Тема 9. Рівнева організація прикладу. | 1 | 12 | 2 | | 4 | 6 | | | | | | | |
| Тема 10. Дизайн рівня представлення. | 1 | 8 | 2 | | | 6 | | | | | | | |
| Тема 11. Дизайн рівня бізнес-логіки. | 1 | 12 | 2 | | 4 | 6 | | | | | | | |
| Тема 12. Дизайн рівня даних. | 1 | 8 | 2 | | | 6 | | | | | | | |
| Тема 13. Дизайн рівня сервісів. | 1 | 12 | 2 | | 4 | 6 | | | | | | | |
| Тема 14. Інтерфейси, взаємодія, еволюція програм і даних | 1 | 10 | 2 | | 2 | 6 | | | | | | | |
| Тема 15. Моделі якості та надійності програмних систем | 1 | 8 | 2 | | | 6 | | | | | | | |
| Разом за модулем 2 | 78 | | 16 | | 14 | 48 | | | | | | | |
| Усього годин | 150 | | 30 | | 30 | 90 | | | | | | | |
| Курсовий проект (робота) | | | | | | | | | | | | | |
| Усього годин | 150 | | 30 | | 30 | 90 | | | | | | | |

3. Теми лекцій

| № з/п | Назва теми | Кількість годин |
|-------|--|-----------------|
| 1 | Вступ до дисципліни «Конструювання програмного забезпечення» | 2 |
| 2 | Введення в конструювання програмного забезпечення. | 2 |
| 3 | Загальні підходи до проектування програмних систем | 2 |
| 4 | Техніка моделювання архітектури ПЗ. | 2 |
| 5 | Концепції сервіс-орієнтованої архітектури ПЗ. | 2 |
| 6 | RESTful та SOAP Web-сервіси. | 2 |
| 7 | Документування архітектури ПЗ. | 2 |
| 8 | Методи об'єктного аналізу і моделювання. | 2 |
| 9 | Рівнева організація прикладу. | 2 |
| 10 | Дизайн рівня представлення. | 2 |
| 11 | Дизайн рівня бізнес-логіки. | 2 |
| 12 | Дизайн рівня даних. | 2 |
| 13 | Дизайн рівня сервісів. | 2 |
| 14 | Інтерфейси, взаємодія, еволюція програм і даних | 2 |
| 15 | Моделі якості та надійності програмних систем | 2 |
| ... | Усього | 30 |

4. Теми лабораторних (практичних, семінарських) занять

| № з/п | Назва теми | Кількість годин |
|-------|--|-----------------|
| 1 | Введення в конструювання програмного забезпечення. | 4 |
| 2 | Загальні підходи до проектування програмних систем | 4 |
| | Концепції сервіс-орієнтованої архітектури ПЗ. | 4 |
| | Документування архітектури ПЗ. | 4 |
| | Рівнева організація прикладу. | 4 |
| | Дизайн рівня бізнес-логіки. | 4 |
| | Дизайн рівня сервісів. | 4 |
| | Інтерфейси, взаємодія, еволюція програм і даних | 2 |
| ... | Усього | 30 |

5. Теми самостійної роботи

| № з/п | Назва теми | Кількість годин |
|-------|--|-----------------|
| 1 | Вступ до дисципліни «Конструювання програмного забезпечення» | 6 |
| 2 | Введення в конструювання програмного забезпечення. | 6 |
| 3 | Загальні підходи до проектування програмних систем | 6 |
| 4 | Техніка моделювання архітектури ПЗ. | 6 |
| 5 | Концепції сервіс-орієнтованої архітектури ПЗ. | 6 |
| 6 | RESTful та SOAP Web-сервіси. | 6 |
| 7 | Документування архітектури ПЗ. | 6 |
| 8 | Методи об'єктного аналізу і моделювання. | 6 |
| 9 | Рівнева організація прикладу. | 6 |
| 10 | Дизайн рівня представлення. | 6 |
| 11 | Дизайн рівня бізнес-логіки. | 6 |
| 12 | Дизайн рівня даних. | 6 |
| 13 | Дизайн рівня сервісів. | 6 |
| 14 | Інтерфейси, взаємодія, еволюція програм і даних | 6 |
| 15 | Моделі якості та надійності програмних систем | 6 |
| ... | Усього | 90 |

6. Методи та засоби діагностики результатів навчання:

(вибрати необхідне чи доповнити)

- екзамен;
- модульні тести;
- захист лабораторних робіт;
- захист курсової роботи

7. Методи навчання *(вибрати необхідне чи доповнити)*:

- лекція (проблемна, інтерактивна);
- лабораторна робота;
- проблемне навчання;
- проєктне навчання (індивідуальне);
- командна робота, мозковий штурм.

8. Оцінювання результатів навчання.

Оцінювання знань здобувача вищої освіти відбувається за 100-бальною шкалою і переводиться в національну оцінку згідно чинного «Положення про екзамени та заліки у НУБіП України»

8.1. Розподіл балів за видами навчальної діяльності

| Вид навчальної діяльності | Результати навчання | Оцінювання |
|--|---|---|
| Модуль 1. Загальні відомості про конструювання програмного забезпечення | | |
| Лабораторна робота 1. | Ознайомитись з призначенням та основними задачами дисципліни Знати ключові принципи та інструменти конструювання ПЗ | 15 |
| Лабораторна робота 2. | | 15 |
| Лабораторна робота 3. | | 15 |
| Лабораторна робота 4. | | 15 |
| Самостійна робота 1. | Знати моделі та методику проектування об'єктно орієнтованих програмних систем. | 10 |
| Модульне тестування 1. | Знати ключові сценарії проектування Знати концепції сервіс-орієнтованої архітектури ПЗ RESTful та SOAP Web сервіси 2/2 Опанувати системи обміну повідомленнями. Опанувати технологію пошарового представлення архітектури | 30 |
| Всього за модулем 1 | | 100 |
| Модуль 2. Керування конструюванням програмного забезпечення | | |
| Лабораторна робота 5. | Опанувати методи об'єктного аналізу і моделювання | 15 |
| Лабораторна робота 6. | | 15 |
| Лабораторна робота 7. | Опанувати технологію рівневої організації прикладу | 15 |
| Лабораторна робота 8. | | 15 |
| Самостійна робота 2. | Опанувати технологію дизайну рівня представлення | 10 |
| Модульне тестування 2. | Опанувати технологію дизайну рівня бізнес-логіки Опанувати технологію дизайну рівня даних Опанувати технологію дизайну рівня сервісів Ознайомитись з принципами створення інтерфейсів Ознайомитись з моделями та метриками якості | 30 |
| Всього за модулем 2 | | 100 |
| Навчальна робота | | $(M1 + M2)/2 * 0,7 \leq 70$ |
| Екзамен/залік | | 30 |
| Всього за курс | | $(\text{Навчальна робота} + \text{екзамен}) \leq 100$ |
| Курсовий проект/робота (за наявності) | | 100 |

8.2. Шкала оцінювання знань здобувача вищої освіти

| | |
|--------------------------------------|--|
| Рейтинг здобувача вищої освіти, бали | Оцінка за національною системою (екзамен/заліки) |
|--------------------------------------|--|

| | |
|--------|--------------|
| 90-100 | відмінно |
| 74-89 | добре |
| 60-73 | задовільно |
| 0-59 | незадовільно |

8.3. Політика оцінювання

| | |
|---|---|
| Політика щодо дедлайнів та перескладання | Терміни виконання робіт визначені в ЕНК. Роботи, які здаються із порушенням термінів без поважних причин, оцінюються на нижчу оцінку. Перескладання модулів відбувається із дозволу лектора за наявності поважних причин (наприклад, лікарняний). |
| Політика щодо академічної доброчесності | Списування під час контрольних робіт та екзаменів заборонені (в т.ч. із використанням мобільних девайсів). Курсові роботи, реферати повинні мати коректні текстові посилання на використану літературу |
| Політика щодо відвідування | Відвідування занять є обов'язковим. За об'єктивних причин (наприклад, хвороба, міжнародне стажування) навчання може відбуватись індивідуально (в он-лайн формі за погодженням із деканом факультету) |

9. Навчально-методичне забезпечення:

- електронний навчальний курс навчальної дисципліни знаходиться за електронною адресою: <https://elearn.nubip.edu.ua/course/view.php?id=2686>
- Бородкін Г.О. Методичні рекомендації до виконання курсової роботи з дисципліни “Конструювання програмного забезпечення” для студентів спеціальності 121 «Інженерія програмного забезпечення» К.: НУБІП України. 2023. - 41 с.

10. Рекомендовані джерела інформації

1. Лавріщева К.М. Програмна інженерія. Підручник –К., «НАНУ» – 2008.–319 с
2. S. McConnell, Code Complete, 2nd ed., Microsoft Press, 2004.
3. D. Budgen, Software Design, 2nd ed., Addison-Wesley, 2003.
4. Guckkenheimer S., Peter J. Software Engineering With Microsoft Visual Studio. Team System. – Adison Wesley, 2006. – 273 p.
5. Бабенко Л.П., Лавріщева К.М. Основи програмної інженерії.– Навч. посібник.–К.: Знання, 2001. –269 с.
6. Лавріщева К.М. Основні напрямки досліджень в програмній інженерії і шляхи їхнього розвитку // Проблеми програмування. – 2003. – № 3–4. – С. 44–58.
7. R.S. Pressman, Software Engineering: A Practitioner’s Approach, McGraw-Hill, 2010.
8. I. Sommerville, Software Engineering, 9th ed., Addison-Wesley, 2011.
9. A. Silberschatz, P.B. Galvin, and G. Gagne, Operating System Concepts, 8th ed., Wiley, 2008.
10. INCOSE, Systems Engineering Handbook: A Guide for System Life Cycle Processes and Activities, version 3.2.2, International Council on Systems Engineering, 2012.
11. S. Friedenthal, A. Morre, and R. Steiner, A Practical Guide to SysML: The Systems Modeling Language, 2nd ed., Morgan Kaufmann, 2012.
12. Допоміжна

13. Zave P., Jackson M. Four Dark Corners of Requirements Engineering // ACM Transactions on Software Engineering, January 1997.– № 1.
14. I. Jacobson, G. Booch, and J. Rumbaugh, The Unified Software Development Process, Addison-Wesley Professional, 1999.
15. J. Nielsen, Usability Engineering, Morgan Kaufmann, 1993.
16. G. Booch, J. Rumbaugh, and I. Jacobson, The Unified Modeling Language User Guide, Addison-Wesley, 1999.
17. E. Gamma et al., Design Patterns: Elements of Reusable Object-Oriented Software, 1st ed., Addison-Wesley Professional, 1994.

Інформаційні ресурси

1. http://uk.wikipedia.org/wiki/Життєвий_цикл_програмного_забезпечення
2. http://uk.wikipedia.org/wiki/Методологія_розробки_програмного_забезпечення
3. http://uk.wikipedia.org/wiki/Гнучка_розробка_програмного_забезпечення
4. http://uk.wikipedia.org/wiki/Програмна_інженерія
5. IEEE Std., 12207-2008 (a.k.a. ISO/IEC 12207:2008) 'Standard for Systems and Software Engineering—Software Life Cycle Processes, IEEE, 2008.
6. IEEE, IEEE Std. 1069-2009 Standard for Information Technology—Systems Design—Software Design Descriptions, IEEE, 2009.
7. ISO/IEC/IEEE., 24765:2010 Systems and Software Engineering—Vocabulary, ISO/IEC/IEEE, 2010.
8. IEEE, "IEEE Std. 1517-2010 Standard for Information Technology—System and Software Life Cycle Processes—Reuse Processes", IEEE, 2010.