

## КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

Тип дисципліни	Обов'язкова
Рівень вищої освіти	Перший
Мова викладання	Українська
Семестр	Сьомий
Обсяг кредитів ЄКТС	6
Форма здобуття освіти	Денна (очна)

**Результати навчання.** Відповідно до Стандарту вищої освіти та освітньої програми дисципліна має забезпечити:

- **компетентності:** здатність застосовувати знання у практичних ситуаціях; здатність працювати в команді; здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування; накопичувати, обробляти та систематизувати професійні знання щодо створення і супроводження програмного забезпечення та визнання важливості навчання протягом всього життя; реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення; здатність здійснювати процес інтеграції системи, застосовувати стандарти і процедури управління змінами для підтримки цілісності, загальної функціональності і надійності програмного забезпечення.

- **програмні результати навчання:** аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки; уміти вибирати та використовувати відповідну задачі методологію створення програмного забезпечення; проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування; вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання; застосовувати на практиці ефективні підходи щодо проектування програмного забезпечення; знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань; застосовувати на практиці інструментальні програмні засоби доменного аналізу, проектування, тестування, візуалізації, вимірювань та документування програмного забезпечення; мати навички командної розробки, погодження, оформлення і випуску всіх видів програмної документації; вміти застосовувати методи компонентної розробки програмного забезпечення; знати та вміти застосовувати методи верифікації та валідації програмного забезпечення.

**Зміст навчальної дисципліни.** Елементи конструювання ПЗ. Ключові принципи конструювання. Інструменти конструювання. Мови опису, представлення, архітектурні каркаси. Техніка моделювання архітектури. Ідентифікація цілей та ключових сценаріїв. Системний підхід до розробки ПЗ. Моделі життєвого циклу ПЗ. Методологія створення ПЗ. Ідентифікація ключових проблем. Конструювання ПЗ та багатозадачність. Захисне конструювання ПЗ, синхронізація процесів, підходи при конструюванні ПЗ. Відладки при конструюванні ПЗ та контроль за ними.

**Запланована навчальна діяльність:** лекцій 34 год., лабораторних занять 34 год., самостійної роботи 112 год.; разом 180 год.

**Методи навчання** методи проблемного викладання, словесні, наочні (лекції); пояснювально-ілюстративні, проблемного викладання, дослідницькі, частково-пошукові (лабораторні заняття), проблемного викладання, дослідницькі, частково-пошукові (самостійна робота: індивідуальні завдання).

**Форми і методи оцінювання результатів навчання:** усне опитування, захист лабораторних робіт. письмові самостійні та контрольні роботи, письмовий іспит

**Форма семестрового контролю:** іспит

**Навчальні ресурси:**

1. Роберт Мартін. Чистий код: Чистий код: створення, аналіз, рефакторинг. - Фабула. 2019 – 416 с.
2. Постіл. С.Д. UML. Уніфікована мова моделювання інформаційних систем: Навч. посіб., 2019. - 321 с.
3. Martin Fowler. Refactoring: Improving the Design of Existing Code (WebEdition), 2nd Edition/ WebEdition.- 2018
4. Aho, Alfred V., Jeffrey D. Ullman. The Theory of Parsing, Translation, and Compiling (Volume 2: Compiling). -Prentice-Hall. -2018
5. Ian Sommerville. Software Engineering, 10th edition. Published by Pearson . July 14th 2021 – 816 p.

**Викладач:** кандидат педагогічних наук, доцент Праворська Н.І.

## SOFTWARE DESIGN

<b>Type of Discipline</b>	Compulsory
<b>Level of Higher Education</b>	First (Bachelor's)
<b>Language of Instruction</b>	Ukrainian, English
<b>Semester</b>	7
<b>ECTS Credits</b>	6
<b>Course study mode</b>	Full-time (Daytime)

**Learning outcomes.** According to the Standard of higher education and the educational program, the discipline must ensure: **competences** : the ability to apply knowledge in practical situations; ability to work in a team; the ability to participate in software design , including modeling (formal description) of its structure, behavior and functioning processes; to accumulate, process and systematize professional knowledge about creating and maintaining software and recognizing the importance of lifelong learning; implement phases and iterations of the life cycle of software systems and information technologies based on relevant models and software development approaches; the ability to carry out the system integration process, apply change management standards and procedures to maintain the integrity, overall functionality and reliability of the software ; Ability to work in parallel on the stages of the software life cycle and effective interaction between the executors of the stages. **program learning outcomes:** to analyze, purposefully search for and choose information and reference resources and knowledge necessary for solving professional tasks, taking into account modern achievements of science and technology; to be able to choose and use the software creation methodology appropriate to the task; conduct a pre-project survey of the subject area, system analysis of the design object; select initial data for design, guided by formal requirements description and modeling methods; apply effective software design approaches in practice; know and apply methods of developing algorithms, designing software and data and knowledge structures; apply in practice instrumental software tools for domain analysis, design, testing, visualization, measurement and software documentation; have skills in team development, approval, design and release of all types of software documentation; be able to apply methods of component software development; know and be able to apply software verification and validation methods . To be able to effectively cooperate with the executors of different stages of the software life cycle, coordinating simultaneous work on the stages. To know and be able to apply techniques, tools and strategies for managing resources, time and communication to ensure effective parallel work on different stages of the software life cycle **Course content.** Elements of software design . Key design principles . Construction tools . Description languages, representations, architectural frameworks. Architecture modeling technique. Identification of goals and key scenarios. A systematic approach to software development. Software life cycle models. Software creation methodology . Identification of key issues. Software design and multitasking. Protective software design, process synchronization, software design approaches. Debugging during software design and their control.

**Planned educational activities:** 34 hours of lectures, 34 hours of laboratory classes, 112 hours of independent work; together 180 hours

**Teaching methods:** methods of problem-based teaching, verbal, visual (lectures); explanatory and illustrative, problem-based teaching, research, partially research-based (laboratory classes), problem-based teaching, research, partially research-based.

**Assessment forms and methods:** oral survey, defense of laboratory works. written independent and control works, written exam

**Semester control form** : exam

**Educational resources:**

1. Robert Martin. Clean Code: Clean Code: Build, Analyze, Refactor . - A story. 2019 – 416 p.
2. The bed S.D. UML. Unified modeling language: Teaching . manual , 2019. - 321 p.
3. Aho , Alfred V. fnd Jeffrey D. Ullman . The Theory of Parsing , Translation , and Compiling ( Volume 2: Compiling ). -Prentice-Hall. -2018
4. Ian Sommerville . Software Engineering , 10th edition . Published by Pearson . July 14th 2021 - 816

**Lecturer:** PhD, Associate Professor Pravorska N.I.