

НАЦІОНАЛЬНИЙ ТРАНСПОРТНИЙ УНІВЕРСИТЕТ  
ФАКУЛЬТЕТ ТРАНСПОРТНИХ ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ  
КАФЕДРА ІНФОРМАЦІЙНИХ СИСТЕМ І ТЕХНОЛОГІЙ

**ЗАТВЕРДЖУЮ:**  
**завідувач кафедри**  
**інформаційних систем і технологій**  
проф. В.В. Гавриленко \_\_\_\_\_  
\_\_\_\_\_ 2020 р.

## СЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

<b>Рівень вищої освіти</b>	перший (бакалаврський)
<b>Спеціальність</b>	121 «Інженерія програмного забезпечення»
<b>Освітня програма</b>	«Інженерія програмного забезпечення»
<b>Вид дисципліни</b>	обов'язкова
<b>Форма навчання</b>	денна
<b>Семестр</b>	6-й семестр навчального плану
<b>Викладач</b>	к.т.н., доцент Лагодіна Людмила Петрівна
<b>Кафедра</b>	інформаційних систем і технологій
	тел. кафедри: +38 (044) 280-70-66
	веб-сайт кафедри: <a href="http://kist.ntu.edu.ua/">http://kist.ntu.edu.ua/</a>
<b>Доступ до матеріалів</b>	Google Classroom: Software Construction
	Syllabus: <a href="http://kist.ntu.edu.ua/nmk_ipz_bak.php">http://kist.ntu.edu.ua/nmk_ipz_bak.php</a> ,
	ел. підручники: <a href="http://kist.ntu.edu.ua/posib_ipz_bak.php">http://kist.ntu.edu.ua/posib_ipz_bak.php</a>
<b>Розклад</b>	<a href="http://www.ntu.edu.ua/studentam/rozklad/">http://www.ntu.edu.ua/studentam/rozklad/</a>
<b>Консультації</b>	<a href="http://kist.ntu.edu.ua/rozkl_kons_kaf_ist.pdf">http://kist.ntu.edu.ua/rozkl_kons_kaf_ist.pdf</a>

### І. АНОТАЦІЯ

**Метою** вивчення дисципліни є отримання здобувачами теоретичних знань і практичних навичок щодо питань, методів та прийомів, пов'язаних із побудовою програмного забезпечення.

**Предметом** вивчення дисципліни є методи проектування та нотації, інструменти реалізації, стандарти та стилі кодування, методи експертного огляду та проблеми супроводження.

Основними **завданнями** вивчення дисципліни є:

- отримання знань щодо фундаментальних основ конструювання програмного забезпечення;
- практика застосування методів рефакторингу для вирішення проблем проектування в кодї;
- практика застосування евристики побудови та супроводження для вдосконалення існуючого коду: способів усунення глобальних змінних та способів тестування складного коду;
- отримання вміння мотивовано обирати мови програмування для розв'язання завдань створення і супроводження програмного забезпечення;
- отримання навичок побудови програмного забезпечення у такий спосіб, щоб воно відповідало цілям, визначеним у проекті, у т.ч. для підприємств транспортної галузі;
- ознайомлення з підходами до модернізації програмного забезпечення: реверсивна інженерія, реінжиніринг, утилізація та реструктуризація;
- ознайомлення зі способами управління конфігурацією, що використовуються у виробничих системах та ін.

**Мова викладання:** українська.

## II. ОПИС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Засвоєння матеріалів лекцій, виконання й захист лабораторних робіт дозволять сформувати знання, уміння й навички, необхідні для успішного працевлаштування й професійної діяльності фахівців, які будуть задіяні у процесі розробки програмного продукту. Наприкінці курсу студенти, зазвичай, *ознайомлені* з фундаментальними складовими конструювання програмного забезпечення, а також *вміють*

- ✓ застосовувати відповідні методи рефакторингу для вирішення проблем проектування в кодї.
- ✓ застосовувати загальні правила побудови і супроводження для поліпшення існуючого коду.
- ✓ реалізовувати нескладний програмний проект, який використовує певний стандарт кодування.
- ✓ будувати програмне забезпечення таким чином, щоб воно відповідало цілям, визначеним у проекті.

### Загальні й професійні компетентності

K01 Здатність до абстрактного мислення, аналізу та синтезу.

K02 Здатність застосовувати знання у практичних ситуаціях.

K15 Здатність розробляти архітектури, модулі та компоненти програмних систем.

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.

K26 Здатність до алгоритмічного та логічного мислення.

K29 Здатність розробляти програмне забезпечення для підвищення якості, безпеки, рівня автоматизації та інтелектуалізації транспортних процесів і систем.

K30 Володіння знаннями про специфіку потреб транспортно-дорожнього комплексу на рівні, достатньому для досягнення мети освітньої програми.

### Програмні результати навчання

PR13 Знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.

PR15 Мотивовано обирати мови програмування та технології розробки для розв'язання завдань створення і супроводження програмного забезпечення.

PR27 Вміти розробляти програмне забезпечення для підвищення якості, безпеки, рівня автоматизації та інтелектуалізації транспортних процесів і систем.

### Зв'язок з іншими дисциплінами

У структурі начального плану курс «Конструювання програмного забезпечення» супроводжує і, відповідно, логічно пов'язаний із навчальними дисциплінами фахового спрямування, що безпосередньо формують компетентності інженера-програміста. Для успішного початку засвоєння матеріалу бажані знання й навички, отримані в рамках наступних навчальних дисциплін: «Архітектура та проектування програмного забезпечення», ОКП4 «Об'єктно-орієнтоване програмування», ОКП3 «Аналіз вимог до програмного забезпечення».

### Розподіл навчальних годин за видами роботи студента на протязі семестру й сесії

<i>Види робіт за навчальним планом</i>	<i>Годин</i>
<i>Аудиторні заняття, у т.ч.:</i>	<b>48</b>
лекції	16
лабораторні роботи	32
<i>Самостійна робота, у т.ч.:</i>	<b>42</b>
підготовка до лабораторних робіт	
підготовка до модульних контрольних робіт	
підготовка до підсумкового контролю	
<i>Загальний обсяг</i>	<b>90</b>
<i>Форма підсумкового контролю</i>	<i>Іспит</i>

**Інструментарій та програмне середовище для виконання лабораторних робіт:** Android Studio (Emulator), Eclipse, SML, Git, Notepad++/ Emacs, P4Merge.

### III. ЗАПЛАНОВАНІ ТЕМИ АУДИТОРНИХ ЗАНЯТЬ

Зміст області знань Software Construction згідно із рекомендаціями IEEE Computer Science (схема 1) та Curriculum Guidelines for Undergraduate Degree Programs in Software Engineering «Software Engineering 2014» (SE 2014) є основою для формування переліку запланованих тем аудиторних занять для навчальної дисципліни «Конструювання програмного забезпечення» (табл.1).



Схема 1. Зміст області знань «Конструювання програмного забезпечення», SWEBOOK IEEE CS, 2014

Таблиця 1. Календарний графік, форма та зміст аудиторних занять.

Графік	Вид і тема аудиторного заняття	Години
<b>Модуль I. Основи конструювання програмного забезпечення</b>		
1 тиждень	<b>Лекція №1. Фундаментальні складові конструювання програмного забезпечення: мінімізація складності, очікування змін, конструювання з можливістю перевірки, стандарти у конструюванні.</b>	2
	Лабораторна робота №1. Установка Android Studio (SDK, Emulator). Складові інтерфейсу програмного середовища. Структура Android Manifest. Тестування рівня знань мови Java.	2
2 тиждень	Лабораторна робота №2. Основи роботи з Activity. Android Context.	2
3 тиждень	<b>Лекція №2. Управління процесом конструювання: моделі конструювання.</b>	2
	Лабораторна робота №3. Елементи інтерфейсу. Атрибути View та ViewGroup.	2
4 тиждень	Лабораторна робота №4. Система зборки Gradle. Можливості Android Debug Bridge. Система контролю версій.	2
5 тиждень	<b>Лекція №3. Планування конструювання. Вимірювання в конструюванні.</b>	2
	Лабораторна робота №5 Фрагменти й файли Preferences.	2
6 тиждень	Лабораторна робота №6. Activity авторизації, Activity профіля. Додавання фрагментів. Toast.	2
<b>Модуль II. Процеси конструювання.</b>		

Графік	Вид і тема аудиторного заняття	Години
7 тиждень	<b>Лекція №4. Проектування в конструюванні. Мови конструювання.</b>	2
	Лабораторна робота №7. Додавання логіки авторизації.	2
8 тиждень	Лабораторна робота №8. Створення персонажів. Захист індивідуальної роботи.	2
9 тиждень	<b>Лекція №5. Кодування. Рефакторинг. Статична перевірка коду.</b>	2
	Лабораторна робота № 9. ML.* Синтаксис vs. семантики vs. ідіоми vs. бібліотек vs. інструментів	2
10 тиждень	Лабораторна робота № 10. Основи ML. Рекурсивні функції та рекурсивні типи. Tail recursion. Тести.	2
11 тиждень	<b>Лекція №6. Тестування в конструюванні. Повторне використання.</b>	2
	Лабораторна робота № 11. ML. Алгебраїчні типи даних, зіставлення зі зразком. Функції вищого порядку. Синтаксичний цукор.	2
12 тиждень	Лабораторна робота № 12. ML. Параметричний поліморфізм та контейнерні типи. Абстрактні типи та модулі.	2
13 тиждень	<b>Лекція №7. Якість конструювання. Інтеграція.</b>	2
	Лабораторна робота №13. Система керування версіями Git.	
14 тиждень	Лабораторна робота №14. Система керування версіями Git.	2
15 тиждень	<b>Лекція №8. Середовища розробки програмного забезпечення</b>	2
	Лабораторна робота №15. Технології й інструменти конструювання програмного забезпечення	2
16 тиждень	Захист лабораторних робіт, модульний контроль (тестування)	2
За семестр		48

Для виконання лабораторної роботи передбачається 4 години на самостійну підготовку студента, для повторення навчального матеріалу на модульну контрольну роботу – 2 години.

#### Рекомендовані учбові матеріали для самостійної підготовки

1. S. McConnell, *Code Complete*, 2nd ed., Microsoft Press, 2004.  
URL: <http://aroma.vn/web/wp-content/uploads/2016/11/code-complete-2nd-edition-v413hav.pdf>
2. Спинеллис Д. *Анализ программного кода на примере проектов OpenSource*, М: Вильямс, 2004. – 524 с.  
URL: <http://padabum.com/d.php?id=42571>
3. Орлов С.А., Цилькер Б.Я. *Технологии разработки программного обеспечения: Учебник для вузов. 4-е изд.* – СПб.: Питер, 2012. – 608 с. Джерело доступу: [https://library.bsuir.by/m/12\\_101945\\_1\\_115155.pdf](https://library.bsuir.by/m/12_101945_1_115155.pdf)
4. V. Meyer, *Object-Oriented Software Construction*, second ed., Prentice Hall, 1997. – 1254 p.  
URL: [https://www.academia.edu/4903777/ObjectOriented\\_Software\\_Construction\\_SECOND\\_EDITION](https://www.academia.edu/4903777/ObjectOriented_Software_Construction_SECOND_EDITION)
5. *Refactoring: Improving the Design of Existing Code*, by Martin Fowler Publisher: Addison-Wesley Professional; 1 edition, 1999 and 2 edition, 2019. URL: [https://www.csie.ntu.edu.tw/~r95004/Refactoring\\_improving\\_the\\_design\\_of\\_existing\\_code.pdf](https://www.csie.ntu.edu.tw/~r95004/Refactoring_improving_the_design_of_existing_code.pdf)  
<http://codecourse.sourceforge.net/materials/Refactoring-Presentation-from-JavaOne.pdf>
6. Michael C. Feathers. *Working Effectively with Legacy Code*, Publisher: Pearson Education, Prentice-Hall – 2005. URL: <http://ptgmedia.pearsoncmg.com/images/9780131177055/samplepages/0131177052.pdf>

#### IV. ПОРЯДОК ТА КРИТЕРІЇ ОЦІНЮВАННЯ

##### Форми контролю

Поточна форма контролю: захист лабораторних робіт, виконання практичних завдань під час аудиторних занять, тести.

Форма модульного контролю: тести.

Підсумкова форма контролю: екзамен.

##### Розподіл балів за формами контролю:

Тести	20%
Лабораторні роботи	40%
Екзамен	40%
	100%

Виконання й захист всіх лабораторних робіт є обов'язковою умовою для допуску студента до екзамену, оскільки формує у студента більшість програмних компетентностей й результатів навчання в рамках цієї дисципліни. Тести дають можливість оцінити знання із переліку наведених вище програмних результатів. Практичні завдання під час аудиторних занять формують здатність працювати у команді. Порядок проведення екзамену включає письмову відповідь студента на теоретичні питання. Порядок оскарження рішення екзаменатора щодо оцінки визначений у «Положенні про освітній процес НТУ».

Якщо максимальна кількість балів за кожне теоретичне питання – 10 балів, тоді:

- за повністю розкритою відповідь на питання студент одержує 10 балів;
- якщо студент дав відповідь на питання, однак допустив незначні помилки, він одержує 7 балів;
- якщо у відповіді не повністю розкрито сутність питання та допущені невірні тлумачення, студент одержує 3 бали;
- якщо студент не надав відповідь на питання, не виконав завдання, або виконав завдання з принциповими помилками, він одержує 0 балів.

Підсумкову семестрову рейтингову оцінку в балах за національною шкалою та шкалою ECTS (табл. 2) вносять до заліково-екзаменаційної відомості, навчальної картки, індивідуального плану та залікової книжки студента.

Таблиця 2. Шкала оцінювання

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою
		для екзамену, курсової роботи
90 – 100	A	відмінно
82-89	B	добре
74-81	C	
64-73	D	задовільно
60-63	E	
35-59	FX	незадовільно з можливістю повторного складання
0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни

## V. ПОЛІТИКА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

**Відвідування** аудиторних занять є обов'язковим (за винятком випадків, коли існує поважна причина, наприклад, хвороба чи дозвіл працівників деканату). Якщо Ви не можете бути присутніми на заняттях, Ви все одно несеєте відповідальність за виконання завдань, що проводились в комп'ютерному класі, а також маєте дізнатись про всі оголошення. Студенти, які мають більше ніж 30% необґрунтованих пропусків, отримують остаточну оцінку, знижену на повний бал (за національною шкалою); студентів, загальна кількість пропусків яких (виправдані та невиправдані) перевищує 80%, з невиконаними лабораторними роботами й незахищеною курсовою роботою на залік не допускають.

Якщо Ви не маєте можливість вчасно показати й захистити індивідуальне завдання під час аудиторних занять, це можна зробити пізніше - під час консультацій, із графіком яких ознайомтесь завчасно на кафедрі інформаційних систем і технологій. **Вчиться організовувати свою роботу** так, щоб уникати фізичного й емоційного перенапруження у випадку виконання й захисту індивідуальних завдань «в останній день».

Можливо, Вам доведеться користуватися ноутбуками протягом певної частини навчального процесу. Під час обговорення практичних завдань, будь ласка, не використовуйте свої ноутбуки, смартфони, планшети чи комп'ютери в аудиторії. Це може відволікати викладача і студентів у групі, а також заважати зосереджуватися на учбовому матеріалі. Якщо Ви використовуєте свій ноутбук чи телефон для аудіо чи відеозапису, необхідно заздалегідь отримати дозвіл викладача. **Повага** один до одного дає можливість ефективніше досягати поставлених командних результатів.

Всі індивідуальні завдання, заплановані в рамках даної дисципліни, студент має виконати самостійно із використанням рекомендованої літератури й отриманих під час аудиторних занять знань та навичок. Цитування в письмових роботах допускається тільки із відповідним посиланням на авторський текст. Рекомендуємо ознайомитись з питаннями **академічної доброчесності** студентів й викладачів, які регламентуються [Положенням про систему забезпечення академічної доброчесності педагогічними, науково-педагогічними, науковими працівниками та здобувачами вищої освіти в Національному транспортному університеті.](#)

Викладач залишає за собою право під час навчального процесу змінювати, за необхідності й з обов'язковим попередженням про це студентів, зміст, графік консультацій, розподіл балів, викладені в цьому силабусі.

1 вересня 2020