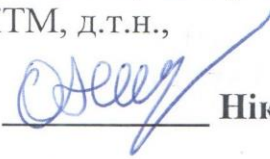


МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ АВТОМОБІЛЬНО-  
ДОРОЖНИЙ УНІВЕРСИТЕТ

Кафедра комп'ютерних технологій і мехатроніки

«ЗАТВЕРДЖУЮ»

Гарант освітньо-професійної  
програми «Програмне забезпечення  
систем» першого (бакалаврського)  
рівня вищої освіти, завідувач  
кафедри КТМ, д.т.н.,

професор  Ніконов  
О.Я.

« 9 » червня 2020р.

СИЛАБУС  
ПРОЕКТУВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО  
ЗАБЕЗПЕЧЕННЯ  
DESIGN AND CONSTRUCTION OF SOFTWARE  
SYLLABUS

освітній ступінь	бакалавр / bachelor
галузь знань	12 Інформаційні технології / Information Technology
спеціальність	121 Інженерія програмного забезпечення / Software Engineering
освітня програма	Програмне забезпечення систем / Systems Software

Харків 2020рік

Автор: Алексієв О.П. професор кафедри комп'ютерних технологій і мехатроніки

Силабус розглянуто та затверджено на засіданні кафедри комп'ютерних технологій та мехатроніки 9.06.2020 р. протокол №18

## **СИЛАБУС**

### **ПРОЕКТУВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

#### **DESIGN AND CONSTRUCTION OF SOFTWARE**

#### **SYLLABUS**

<b>освітній ступінь</b>	<b>бакалавр / bachelor</b>
<b>галузь знань</b>	<b>12 Інформаційні технології / Information Technology</b>
<b>спеціальність</b>	<b>121 Інженерія програмного забезпечення / Software Engineering</b>
<b>освітня програма</b>	<b>Програмне забезпечення систем / Systems Software</b>

## **Анотація курсу**

### **1. Викладачі:**

**Лектор** Алексієв Олег Павлович,

- Д.т.н., професор кафедри комп'ютерних технологій та мехатроніки;
- Педагогічний стаж – 47 років;
- Контактний телефон : 057-707-37-43;
- e-mail: o.p.alex@gmail.com;
- наукові інтереси: машинне навчання, штучний інтелект;
- стажування та підвищення кваліфікації.

**2. Дисципліна:** Проектування та конструювання програмного забезпечення

- рік навчання 3
- семестр навчання 5
- кількість кредитів 4
- кількість годин за семестр:
  - Лекційних 16
  - Лабораторних 32
  - На самостійне опрацювання 72
- кількість аудиторних годин на тиждень
  - лекційних 1
  - лабораторних 2

### **3. Час та місце проведення:**

Аудиторні заняття – відповідно до розкладу ХНАДУ з урахуванням специфіки дисципліни проведення дисципліни передбачено в аудиторіях 123, 214, 313;

Позааудиторна робота – самостійна робота студента

### **4. Пререквизити та постреквизити навчальної дисципліни:**

- **Пререквизити:** «Алгоритмізація та програмування», «Об'єктно-орієнтоване програмування», «Проектування програмного забезпечення», «Аналіз вимог до програмного забезпечення».
- **Постреквизити:** «Тестування програмного забезпечення», «Програмування баз даних».

### **5. Характеристика дисципліни:**

**5.1 Призначення навчальної дисципліни:** Навчальна дисципліна «Проектування та конструювання програмного забезпечення» є загальнодисципліною і входить в професійний цикл, який формує базовий рівень знань для освоєння загально-професійних навичок.

**5.2. Мета вивчення навчальної дисципліни** «Проектування та конструювання програмного забезпечення» є підготовка спеціаліста який володіє основними навичками створення програмного забезпечення відповідно до створених проектів з використанням інтегрованих систем розробки програмного забезпечення.

**5.3 Задачі вивчення дисципліни:** По завершенні вивчення дисципліни студенти повинні:

- уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення;
- проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування;
- знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань;
- вміти застосовувати методи компонентної розробки програмного забезпечення;
- знати та вміти застосовувати методи верифікації та валідації програмного забезпечення.

**5.4 Зміст навчальної дисципліни:** відповідає навчальній та робочій програмі, яка відповідає запитам роботодавців

### 5.5 План вивчення дисципліни

Результати навчання	Навчальна діяльність	Робочий час студента (год.)	Оцінювання (бал)
<b>Загальні та спеціальні компетентності:</b> здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення; здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів	Тема 1. Введення в конструювання програмного забезпечення <i>План лекції:</i> <ul style="list-style-type: none"> <li>- основні задачі проектування та конструювання ПЗ;</li> <li>- каскадна модель життєвого циклу ПЗ;</li> <li>- методи проектування ПЗ</li> </ul> <i>Список рекомендованих джерел:</i> <i>Основний:</i> 1,2,3,4,5 <i>Додатковий:</i> 1,2 <i>Інтернет-ресурси:</i>	8	8
	<b>Практичне заняття 1.</b> Розробка технічного завдання на проектування програми <i>Мета роботи:</i> Отримати практичні навички розробки вимог до ПЗ <i>Завдання:</i> Сформулювати вимоги до ПЗ		

<p>функціонування.  <b>Результати навчання:</b>  проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування.</p>	<p><b>Завдання для самостійної роботи.</b>  Самостійне опрацювання літературних джерел, які зазначені у списку  <i>Питання, винесені на самостійне опрацювання:</i>  Процес формування технічного завдання</p>		
<p><b>Загальні та спеціальні компетентності:</b>  здатність розробляти архітектури, модулі та компоненти програмних систем  <b>Результати навчання:</b>  уміння вибирати та використовувати відповідну задачу методологію створення програмного забезпечення</p>	<p>Тема 2. Сучасні принципи проектування архітектури ПЗ  <i>План лекції:</i></p> <ul style="list-style-type: none"> <li>- уніфікована мова моделювання;</li> <li>- статичні моделі об'єктно-орієнтованих програмних систем;</li> <li>- динамічні моделі об'єктно-орієнтованих програмних систем</li> </ul> <p><b>Список рекомендованих джерел:</b>  <i>Основний: 1,2</i>  <i>Додатковий: 1,2,3</i>  <i>Інтернет-ресурси:</i></p>	12	10
	<p><b>Практичне заняття 2.</b> Діаграма класів  <i>Мета роботи:</i>  Отримати практичні навички побудови діаграм класів  <i>Завдання:</i>  Побудувати діаграму класів</p>		
	<p><b>Завдання для самостійної роботи.</b>  Самостійне опрацювання літературних джерел, які зазначені у списку  <i>Питання, винесені на самостійне опрацювання:</i>  Використання діаграм UML при проектуванні та конструюванні ПЗ</p>		
<p><b>Загальні та спеціальні компетентності:</b>  здатність розробляти архітектури, модулі та компоненти програмних систем  <b>Результати навчання:</b>  проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування;  знати і застосовувати</p>	<p>Тема 3. Метрики об'єктно-орієнтованих програмних систем  <i>План лекції:</i></p> <ul style="list-style-type: none"> <li>- метричні особливості об'єктно-орієнтованих програмних систем;</li> <li>- мери зв'язку для об'єктно-орієнтованих програмних систем;</li> <li>- набір метрик Чідамбера і Кемерера;</li> <li>- метрики Лоренца і Кідда;</li> </ul> <p><b>Список рекомендованих джерел:</b>  <i>Основний: 1,4,5</i>  <i>Додатковий: 2</i>  <i>Інтернет-ресурси:</i></p>	16	15

<p>методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань; вміти застосовувати методи компонентної розробки програмного забезпечення; знати та вміти застосовувати методи верифікації та валідації програмного забезпечення</p>	<p><b>Практичне заняття 3. Розрахунок складності програмної системи</b>  <i>Мета роботи:</i>  Отримати практичні навички аналізу програмних систем на зв'язність та зчеплення  <i>Завдання:</i>  Провести аналіз програмної системи на зв'язність та зчеплення за допомогою метрики Чідамбера і Кемерера</p>		
<p><b>Загальні та спеціальні компетентності:</b>  здатність розробляти архітектури, модулі та компоненти програмних систем  <b>Результати навчання:</b>  уміння вибирати та використовувати відповідну задачу методологію створення програмного забезпечення</p>	<p>Тема 4. Загальні концепції конструювання  <i>План лекції:</i></p> <ul style="list-style-type: none"> <li>- основні принципи конструювання;</li> <li>- модульність;</li> <li>- багаторівневі додатки;</li> </ul> <p><b>Список рекомендованих джерел:</b>  <i>Основний:</i> 1,2,3  <i>Додатковий:</i> 2,3  <i>Інтернет-ресурси:</i></p>	14	12
	<p><b>Практичне заняття 4. Оцінка системи на зчеплення та зв'язність</b>  <i>Мета роботи:</i>  Отримати практичний досвід аналізу систем на зчеплення та зв'язність  <i>Завдання:</i>  Проаналізувати програмну систему на зв'язність і зчеплення</p>		
	<p><b>Завдання для самостійної роботи.</b>  Самостійне опрацювання літературних джерел, які зазначені у списку  <i>Питання, винесені на самостійне опрацювання:</i>  Рефракторинг</p>		

<p><b>Загальні та спеціальні компетентності:</b> здатність розробляти архітектури, модулі та компоненти програмних систем; здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення</p> <p><b>Результати навчання:</b> знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань</p>	<p>Тема 5. Об'єктно-орієнтоване програмування. Принципи S.O.L.I.D <i>План лекції:</i></p> <ul style="list-style-type: none"> <li>- застосування принципів S.O.L.I.D при конструюванні ПЗ;</li> <li>- принцип єдиної відповідальності;</li> <li>- принцип відкритості / закритості;</li> <li>- принцип підстановки Лісков</li> <li>- принцип поділу інтерфейсів</li> <li>- принцип інверсії залежностей;</li> </ul> <p><b>Список рекомендованих джерел:</b> <i>Основний: 1, 3</i> <i>Додатковий: 1,2</i> <i>Інтернет-ресурси:</i></p> <p><b>Практичне заняття 5.</b> Принципи S.O.L.I.D. при конструюванні програмного забезпечення. <i>Мета роботи:</i> Отримати практичні навички застосування принципів S.O.L.I.D. при створенні ПЗ <i>Завдання:</i> Проаналізувати код на відповідність принципам S.O.L.I.D</p> <p><b>Завдання для самостійної роботи.</b> Самостійне опрацювання літературних джерел, які зазначені у списку <i>Питання, винесені на самостійне опрацювання:</i> Огляд узагальнених шаблонів проектування</p>	16	15
<p><b>Загальні та спеціальні компетентності:</b> здатність розробляти архітектури, модулі та компоненти програмних систем; здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення</p>	<p>Тема 6. Шаблони проектування. Породжуючі шаблони <i>План лекції:</i></p> <ul style="list-style-type: none"> <li>- шаблон абстрактна фабрика;</li> <li>- шаблон будівельник;</li> <li>- шаблон фабричний метод;</li> <li>- шаблон одиночка</li> </ul> <p><b>Список рекомендованих джерел:</b> <i>Основний: 1,3</i> <i>Додатковий: 1,2</i> <i>Інтернет-ресурси:</i></p>	16	15

<p><b>Результати навчання:</b> знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань; вміти застосовувати методи компонентної розробки програмного забезпечення</p>	<p><b>Практичне заняття 6.</b> Вибір породжуючого шаблону проектування і його реалізація в програмному кодї  <b>Мета роботи:</b>  Отримати практичний досвід застосування породжуючих шаблонів проектування  <b>Завдання:</b>  Вирішити завдання за допомогою породжуючого шаблону проектування</p>		
<p><b>Загальні та спеціальні компетентності:</b> здатність розробляти архітектуру, модулі та компоненти програмних систем; здатність обґрунтовано обирати та освоювати інструментарій з розробки та супроводження програмного забезпечення</p>	<p>Тема 7. Шаблони проектування. Структурні і шаблони поведінки  <b>План лекції:</b>  <ul style="list-style-type: none"> <li>- структурні шаблони;</li> <li>- поведінкові шаблони;</li> </ul> <b>Список рекомендованих джерел:</b>  <b>Основний:</b> 1, 3  <b>Додатковий:</b> 1,2,3  <b>Інтернет-ресурси</b></p>		
<p><b>Результати навчання:</b> знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань; вміти застосовувати методи компонентної розробки програмного забезпечення</p>	<p><b>Практичне заняття 7.</b> Структурні і шаблони поведінки  <b>Мета роботи:</b>  ознайомитися з структурними і поведінковими шаблонами проектування, навчитися застосовувати їх при проектуванні і розробці ПЗ.  <b>Завдання:</b>  Вирішити завдання за допомогою шаблону проектування</p>	16	15
	<p><b>Завдання для самостійної роботи.</b>  Самостійне опрацювання літературних джерел, які зазначені у списку  <b>Питання, винесені на самостійне опрацювання:</b>  Огляд структурних шаблонів проектування. Огляд поведінкових шаблонів проектування</p>		



<b>Загальні та спеціальні компетентності:</b> здатність реалізовувати фази та ітерації життєвого циклу програмних систем та інформаційних технологій на основі відповідних моделей і підходів розробки програмного забезпечення <b>Результати навчання:</b> знати та вміти застосовувати методи верифікації та валідації програмного забезпечення	Тема 8. Введення в тестування програмного забезпечення <i>План лекції:</i> <ul style="list-style-type: none"> <li>- продукти, що піддаються тестуванню;</li> <li>- процес тестування;</li> <li>- якість ПЗ;</li> <li>- методи тестування</li> </ul> <b>Список рекомендованих джерел:</b> <i>Основний: 1, 3</i> <i>Додатковий: 1,3</i> <i>Інтернет-ресурси</i>	10	10	
	<b>Практичне заняття 8.</b> Тестування за допомогою білого та чорного ящиків <i>Мета роботи:</i> Отримання практичного досвіду зі складання тестових програм <i>Завдання:</i> Розробити тестові умови для індивідуального завдання			
	<b>Завдання для самостійної роботи.</b> Самостійне опрацювання літературних джерел, які зазначені у списку <i>Питання, винесені на самостійне опрацювання:</i> Огляд методик тестування ПЗ			
<b>Разом</b>		<b>120/4</b>	<b>100</b>	
<b>Підсумковий контроль</b>		<b>Письмовий екзамен</b>		

## 6. Рекомендовані джерела інформації

### Основна

1. Макконнелл, Стив Совершенный код : практическое руководство по разработке программного обеспечения : пер. с англ. / С. Макконнелл. — Москва: Русская редакция, 2013. — 869 с.
2. Брауде, Эрик Дж. Технология разработки программного обеспечения / Э. Д. Брауде. — СПб.: Питер, 2004. — 655 с.
3. Орлов, Сергей Александрович Технологии разработки программного обеспечения. Современный курс по программной инженерии : учебник / С. А. Орлов, Б. Я. Цилькер. — 4-е изд.. — СПб.: Питер, 2012. — 608 с.

4. Рамбо, Дж. UML 2.0. Объектно-ориентированное моделирование и разработка : пер. с англ. / Дж. Рамбо, М. Блаха. — 2-е изд. — СПб.: Питер Пресс, 2007. — 544 с.

5. Буч Г. Язык UML. Руководство пользователя [http://e.lanbook.com/books/element.php?pl1\\_id=1246](http://e.lanbook.com/books/element.php?pl1_id=1246)

### Допоміжна

1. Гагарина Л.Г. Технология разработки программного обеспечения: учебное пособие / Л. Г. Гагарина, Е. В. Кокорева, Б. Д. Виснадул. — Москва: Форум Инфра-М, 2013. — 400 с.

2. Басс Л. Архитектура программного обеспечения на практике / Л. Басс, П. Клементс, Р. Кацман. — 2-е изд. — СПб.: Питер, 2006.

3. Круз Р.Л. Структуры данных и проектирование программ: пер. с англ. / Р. Л. Круз. — Москва: БИНОМ. Лаборатория знаний, 2008. — 765 с.

### 7. Контроль та оцінювання результатів навчання:

Під час вивчення дисципліни «Проектування та конструювання програмного забезпечення» викладачем здійснюється поточний та підсумковий контроль, що передбачає:

- перевірку рівня засвоєння теоретичного матеріалу (усне та письмове опитування);
- захист лабораторних робіт (протягом семестру)

**8. Критерії оцінювання результатів навчання** Підсумковий контроль знань та компетентностей студентів з навчальної дисципліни здійснюється на підставі заліку.

Відповідність підсумкової семестрової рейтингової оцінки в балах оцінці за національною шкалою та шкалою ECTS:

Оцінка в балах	Оцінка за національною шкалою	Оцінка за шкалою ECTS	
		Оцінка	Пояснення
90-100	Відмінно	A	Відмінно (відмінне виконання лише з незначною кількістю помилок)
82 – 89	Добре	B	Дуже добре (вище середнього рівня з кількома помилками)
75 – 81		C	Добре (в загальному вірне виконання з певною кількістю суттєвих помилок)
67 – 74	Задовільно	D	Задовільно (непогано, але зі значною кількістю недоліків)
60 – 66		E	Достатньо (виконання задовольняє мінімальним критеріям)
35 – 59	Незадовільно	FX	Незадовільно (з можливістю повторного складання)
1 – 34		F	Незадовільно (з обов'язковим повторним курсом)