

Міністерство освіти і науки України  
Національний аерокосмічний університет ім. М. Є. Жуковського  
«Харківський авіаційний інститут»

Кафедра математичного моделювання та штучного інтелекту (№ 304)

**ЗАТВЕРДЖУЮ**

Голова НМК 2

  
(підпис)

Д. М. Крицький  
(ініціали та прізвище)

« 31 » серпня 2021 р.

**СИЛАБУС ВИБІРКОВОЇ  
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Сучасні парадигми програмування складних систем  
(назва навчальної дисципліни)

**Галузь знань:** 10 Природничі науки, 11 Математика та статистика, 12 Інформаційні технології, 15 Автоматизація та приладобудування, 16 Хімічна та біоінженерія, 17 Електроніка та телекомунікації, 19 Архітектура та будівництво

**Спеціальність:** 101 Екологія, 103 Науки про Землю, 113 Прикладна математика, 121 Інженерія програмного забезпечення, 122 Комп'ютерні науки, 123 Комп'ютерна інженерія, 124 Системний аналіз, 125 Кібербезпека, 151 Автоматизація та комп'ютерно-інтегровані технології, 152 Метрологія та інформаційно-вимірювальна техніка, 163 Біомедична інженерія, 172 Телекомунікації та радіотехніка, 173 Авіоніка, 193 Геодезія та землеустрій

**Освітня програма:** усі освітні програми за відповідними спеціальностями

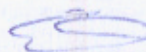
**Рівень вищої освіти:** *другий (магістерський)*

**Силабус введено в дію з 01.09.2021 року**

**Харків – 2021 р.**



Розробник: Меняйлов Є. С., ст. викл.  
(прізвище та ініціали, посада, науковий ступінь і вчене звання)

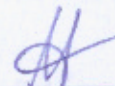
  
(підпис)

Силабус навчальної дисципліни розглянуто на засіданні кафедри  
математичного моделювання та штучного інтелекту  
(назва кафедри)

Протокол № 2 від « 27 » 08 2021 р.

Завідувач кафедри

Д.Т.Н. доцент  
(науковий ступінь та вчене звання)

  
(підпис)

Андрій ЧУХРАЙ  
(ім'я та прізвище)

Погоджено з представником здобувачів освіти:

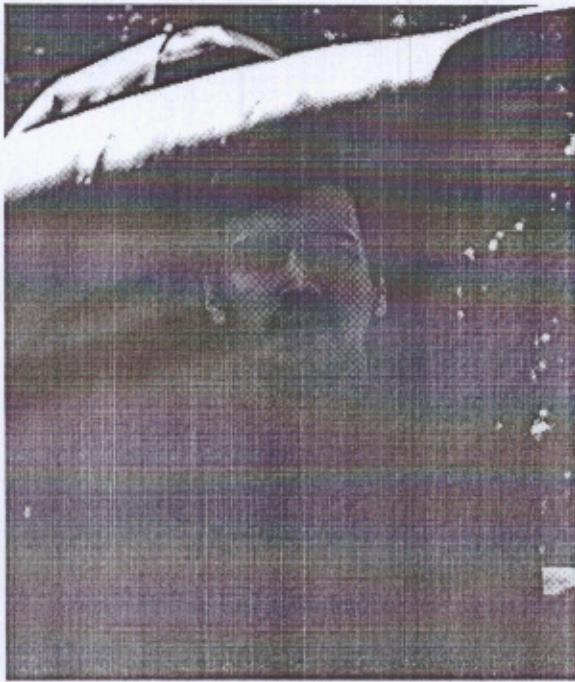
Бюджету студентською самоврядування  
факультету №3

  
(підпис)

А.І.Блашов  
(ім'я та прізвище)



## 1. Загальна інформація про викладача



Меняйлов Євген Сергійович. З 2015 року викладає в університеті наступні дисципліни:

- архітектура обчислювальних систем;
- проектування програмних систем;
- інтелектуальні системи;
- аналіз даних.

Напрями наукових досліджень:

1. Математичні моделі і методи розв'язання задач вдосконалення технічних систем в умовах невизначеності вхідних даних.

2. Інтелектуальні моделі, методи та інформаційні технології для дослідження статистичних даних о пацієнтах в медичній діагностиці.

## 2. Опис навчальної дисципліни

**Семестр, в якому викладається дисципліна** – 1 семестр.

**Обсяг дисципліни:** 5 кредитів ЄКТС / 150 годин, у тому числі аудиторних – 64 год., самостійної роботи здобувачів – 86 год.

**Форма здобуття освіти** – денна, дистанційна, дуальна.

**Дисципліна** вибіркова.

**Види навчальної діяльності** – лекції, практичні (лабораторні) заняття

**Види контролю** – модульний контроль, іспит.

**Мова викладання** – українська.

**Пререквізити** – програмування, знання з об'єктно орієнтованого програмування, володіння навичками створення та користування діаграмами на мові UML.

**Кореквізити** – інтелектуальні системи, математичне моделювання, інтелектуальні програмні комплекси, системи та методи прийняття рішень, написання магістерської роботи.



### **3. Мета та завдання навчальної дисципліни**

**Мета:** вивчення навчальної дисципліни «Сучасні парадигми програмування складних систем» полягає в формуванні системи знань, способів діяльності та творчих здібностей з основних теоретичних положень про підходи до архітектури складних систем через використання патернів проектування та використання різних парадигм програмування, а також засвоєння вмінь які б дозволяли на практиці реалізувати ці знання

**Завдання:** засвоєння знань про парадигми програмування при створенні програмних систем; вивчення підходів до створення елементів складних систем як частини загальної архітектури програмного продукту, вивчення патернів проектування особливостей їх використання при програмуванні складних систем.

#### **Компетентності, які набуваються:**

- здатність розроблювати елементи програмної архітектури сучасних інформаційних систем;
- здатність розроблювати функціональні модулі програмних систем через їх «поведінку» або «структуру»;
- здатність використовувати уніфіковану мову проектування UML для опису загальної архітектури складних систем, або відповідних її елементів;
- здатність розроблювати програмне забезпечення відповідно до обраної моделі життєвого циклу інформаційних систем;
- здатність використовувати різні патерни (шаблонів) проектування та можливостями їх використання при розробці архітектури інформаційних систем;
- здатність використовувати особливості програмування компонентів архітектури програмних систем, відповідно до обраної парадигми програмування.

#### **Очікувані результати навчання:**

- вміти проектувати архітектуру та програмувати компоненти сучасних складних інформаційних систем;
- обирати належні засоби для розробки або дослідження (середовище розробки, мова програмування, програмне забезпечення та програмні пакети тощо), що дозволяють знайти правильне і ефективне рішення;
- мати уявлення про види патернів проектування програмних систем, та практичні аспекти їх використання;
- знати особливості розборки функціональності і поведінки складної системи;
- знати особливості розборки статичної структури складної системи;



#### **4. Зміст навчальної дисципліни**

**Змістовний модуль 1. Проектування системи через її поведінкові аспекти або через її структурні елементи.**

- *Обсяг самостійної роботи здобувачів: 12-16 годин – самостійне опрацювання матеріалу за темами..*

Діаграми: комунікації, огляду взаємодії, синхронізації для проектування складних систем.

Діаграми: композитної (складової структури), кооперації, об'єктів, пакетів, профілів для проектування складних систем.

**Тема 1. UML як уніфікована мова опису складних інформаційних систем.**

- *Форма занять: лекція, самостійна робота.*

- *Обсяг аудиторного навантаження: 1-2 години.*

Об'єктне моделювання в області розробки програмного забезпечення, моделювання бізнес-процесів, системне проектування та відображення організаційних структур.

- *Обсяг самостійної роботи здобувачів: 1-2 години.*

Опрацювання матеріалу лекцій. Формування питань до викладача. Оформлення практичної (лабораторної роботи) та підготовка до її здачі.

**Тема 2. Проектування варіантів використання системи і взаємодії програмних модулів (або компонентів).**

- *Форма занять: лекція, практична (лабораторна) робота, самостійна робота.*

- *Обсяг аудиторного навантаження: 6-8 годин.*

- *Практична (лабораторна) робота: «Проектування варіантів використання системи».*

- *Практична (лабораторна) робота: «Проектування взаємодії в системі».*

Розробка функціональності і поведінки системи через відносини, що існують між діючими особами і варіантами використання системи.

Погляд на систему через взаємодію об'єктів, впорядкованих за часом їх прояву та послідовність повідомлень, якими вони обмінюються.

- *Обсяг самостійної роботи здобувачів: 2-4 години.*

Опрацювання матеріалу лекцій. Формування питань до викладача. Оформлення практичних (лабораторних робіт) та підготовка до їх здачі.

**Тема 3. Проектування діяльності і станів в середині системи.**

- *Форма занять: лекція, практична (лабораторна) робота, самостійна робота.*

- *Обсяг аудиторного навантаження: 6-8 годин.*



- Практична (лабораторна) робота: «Проектування діяльності компонентів системи».

- Практична (лабораторна) робота: «Проектування станів компонентів системи».

Специфікація поведінки системи у вигляді координованого послідовного і паралельного виконання підлеглих елементів – вкладених видів діяльності і окремих дій, з'єднаних між собою потоками, які йдуть від виходів одного вузла до входів іншого. Проектування специфікації послідовності станів, через які проходить об'єкт або взаємодія у відповідь на події свого життя, а також відповідні дії об'єкта на ці події.

- Обсяг самостійної роботи здобувачів: 2-4 години.

Опрацювання матеріалу лекцій. Формування питань до викладача. Оформлення практичних (лабораторних робіт) та підготовка до їх здачі.

**Тема 4. Класи системи, атрибути, методи і залежності між класами.**

- Форма занять: лекція, практична (лабораторна) робота, самостійна робота.

- Обсяг аудиторного навантаження: 4-6 годин.

- Практична (лабораторна) робота: «Проектування класів системи».

Статичний опис структури системи. Точки зору на структуру системи в залежності від цілей і застосування: концептуальна точка зору, точка зору специфікації, точка зору реалізації.

- Обсяг самостійної роботи здобувачів: 1-2 години.

Опрацювання матеріалу лекцій. Формування питань до викладача. Оформлення практичної (лабораторної роботи) та підготовка до її здачі.

**Тема 5. Програмні та фізичні компоненти системи.**

- Форма занять: лекція, практична (лабораторна) робота, самостійна робота.

- Обсяг аудиторного навантаження: 4-6 годин.

- Практична (лабораторна) робота: «Проектування компонентів системи».

Розбиття програмної системи на структурні компоненти (програмні та фізичні) та опис зв'язку між компонентами.

- Обсяг самостійної роботи здобувачів: 1-2 години.

Опрацювання матеріалу лекцій. Формування питань до викладача. Оформлення практичної (лабораторної роботи) та підготовка до її здачі.

**Тема 6. Розробка діаграм розгортання системи.**

- Форма занять: лекція, практична (лабораторна) робота, самостійна робота.

- Обсяг аудиторного навантаження: 4-6 годин.

- Практична (лабораторна) робота: «Моделювання вузлів системи».



Моделювання працюючих вузлів (апаратних засобів) і артефактів, розгорнутих на них у вигляді діаграм розгортання.

- *Обсяг самостійної роботи здобувачів: 1-2 години.*

Опрацювання матеріалу лекцій. Формування питань до викладача. Оформлення практичної (лабораторної роботи) та підготовка до її здачі.

#### **Модульний контроль**

- *Форма занять: написання модульної роботи в аудиторії (за рішенням лектора допускається проведення у дистанційній формі).*

- *Обсяг аудиторного навантаження: 2 години*

- *Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): відсутні.*

- *Обсяг самостійної роботи здобувачів – за необхідністю. Підготовка до модульного контролю.*

**Змістовний модуль 2. Парадигми програмування та патерни проектування.**

- *Обсяг самостійної роботи здобувачів: 12-16 годин – самостійне опрацювання матеріалу за темами.*

Шаблони GOF: основні шаблони, твірні шаблони, структурні шаблони, шаблони поведінки, шаблони паралельних операцій. Шаблони GRASP: інформаційний експерт, творець примірників класу, низька зв'язаність, високе зчеплення, поліморфізм, штучний, перенаправлення, стійкий до змін.

#### **Тема 1. Парадигми програмування.**

- *Форма занять: лекція, самостійна робота.*

- *Обсяг аудиторного навантаження: 4-6 годин.*

Парадигма програмування, як спосіб концептуалізації, що визначає організацію роботи програмного продукту і структурування програмної архітектури. Основні парадигми програмування: процедурне програмування, об'єктно-орієнтоване програмування, функціональне програмування, імперативне програмування, декларативне програмування, прототипне програмування, аспектно-орієнтоване програмування, предметно-орієнтоване програмування, функціонально-орієнтоване програмування, структурне програмування, модульне програмування, збірне програмування, програмування з абстрактними типами даних, схемне програмування, логічне програмування, паралельне програмування, компонентне програмування, агентно-орієнтоване програмування, алгебраїчне програмування.

- *Обсяг самостійної роботи здобувачів: 1-2 години.*

Опрацювання матеріалу лекцій. Формування питань до викладача. Оформлення практичної (лабораторної роботи) та підготовка до її здачі.



## **Тема 2. Породжуючи патерни.**

- Форма занять: лекція, практична (лабораторна) робота, самостійна робота.

- Обсяг аудиторного навантаження: 4-6 годин.

- Практична (лабораторна) робота: «Використання породжуючих патернів у складних системах».

Гнучке створення об'єктів системи без внесення зайвих залежностей.

- Обсяг самостійної роботи здобувачів: 1-2 години.

Опрацювання матеріалу лекцій. Формування питань до викладача. Оформлення практичної (лабораторної роботи) та підготовка до її здачі.

## **Тема 3. Структурні патерни.**

- Форма занять: лекція, практична (лабораторна) робота, самостійна робота.

- Обсяг аудиторного навантаження: 4-6 годин.

- Практична (лабораторна) робота: «Використання структурних патернів у складних системах».

Моделювання різноманітних способів побудови зв'язків між об'єктами в системі.

- Обсяг самостійної роботи здобувачів: 1-2 години.

Опрацювання матеріалу лекцій. Формування питань до викладача. Оформлення практичної (лабораторної роботи) та підготовка до її здачі.

## **Тема 4. Поведінкові патерни.**

- Форма занять: лекція, практична (лабораторна) робота, самостійна робота.

- Обсяг аудиторного навантаження: 4-6 годин.

- Практична (лабораторна) робота: «Використання поведінкових патернів у складних системах».

Створення ефективної комунікації між об'єктами

- Обсяг самостійної роботи здобувачів: 1-2 години.

Опрацювання матеріалу лекцій. Формування питань до викладача. Оформлення практичної (лабораторної роботи) та підготовка до її здачі.

## **Модульний контроль.**

- Форма занять: написання модульної роботи в аудиторії (за рішенням лектора допускається проведення у дистанційній формі).

- Обсяг аудиторного навантаження: 2 години

- Обов'язкові предмети та засоби (обладнання, устаткування, матеріали, інструменти): відсутні.



– *Обсяг самостійної роботи здобувачів* – за необхідністю. Підготовка до модульного контролю.

### **5. Індивідуальні завдання**

Не передбачено навчальним планом

### **6. Методи навчання**

Словесні, наочні, практичні.

### **7. Методи контролю**

Поточний контроль (теоретичне опитування й розв'язання практичних завдань), модульний контроль (тестування за розділами курсу) та підсумковий (семестровий) контроль (іспит).

### **8. Критерії оцінювання та розподіл балів, які отримують здобувачі**

| Складові навчальної роботи                         | Бали за одне заняття (завдання) | Кількість занять (завдань) | Сумарна кількість балів |
|----------------------------------------------------|---------------------------------|----------------------------|-------------------------|
| <b>Змістовний модуль 1</b>                         |                                 |                            |                         |
| Робота на лекціях                                  | 0...2                           | 5                          | 0...10                  |
| Виконання і захист лабораторних (практичних) робіт | 0...5                           | 7                          | 0...35                  |
| Модульний контроль                                 | 0...15                          | 1                          | 0...20                  |
| <b>Змістовний модуль 2</b>                         |                                 |                            |                         |
| Робота на лекціях                                  | 0...2                           | 5                          | 0...10                  |
| Виконання і захист лабораторних (практичних) робіт | 0...5                           | 3                          | 0...15                  |
| Модульний контроль                                 | 0...15                          | 1                          | 0...20                  |
| <b>Усього за семестр</b>                           |                                 |                            | <b>0...100</b>          |

Іспит проводиться у вигляді тестування та виконання практичних завдань. Тест складається з 20 питань закритого типу (за правильну відповідь на одне питання здобувач отримує 3 бала) та двох практичних завдань (максимальна кількість балів за практичне завдання – 20).

Під час складання семестрового іспиту здобувач має можливість отримати максимум 100 балів.

### **Критерії оцінювання роботи здобувача протягом семестру**

*Задовільно (60-74)* – мати мінімум знань і умінь для забезпечення програмних результатів навчання. Відпрацювати та захистити всі практичні роботи.



*Добре (75-89)* – знати основні теми дисципліни. Достатньо знати основні етапи створення складних систем, особливості реалізації компонентів складних систем, описувати програмну архітектуру. Відпрацювати та захистити всі практичні роботи.

*Відмінно (90-100)* – мати знання, що дозволять самостійно, вільно та обґрунтовано відповідати на будь які питання щодо програмної реалізації архітектури складної системи (та її компонентів), відповідно до обраної парадигми програмування та стеку технологій. Відпрацювати та захистити всі лабораторні роботи.

### Шкала оцінювання: бальна і традиційна

| Сума балів | Оцінка за традиційною шкалою  |               |
|------------|-------------------------------|---------------|
|            | Іспит, диференційований залік | Залік         |
| 90 – 100   | Відмінно                      | Зараховано    |
| 75 – 89    | Добре                         |               |
| 60 – 74    | Задовільно                    |               |
| 0 – 59     | Незадовільно                  | Не зараховано |

### 9. Політика навчального курсу

Відпрацювання пропущених занять відбувається відповідно до розкладу консультацій, за попереднім погодженням з викладачем. Питання, що стосуються академічної доброчесності, розглядає викладач або за процедурою, визначеною у Положенні про академічну доброчесність.

### 10. Методичне забезпечення

Не має.

### 11. Рекомендована література

#### Базова

1. Peter Van Roy. "Programming Paradigms: What Every Programmer Should Know" (PDF). [info.ucl.ac.be](http://info.ucl.ac.be). Retrieved 2021-08-27.

2. About the Unified Modeling Language Specification Version 2.5.1. [www.omg.org](http://www.omg.org). Retrieved 2021-08-27.

3. Unified Modeling Language User Guide, The (2 ed.). Addison-Wesley. 2005. p. 496. ISBN 0321267974. , See the sample content, look for history/

4. Manolescu, Dragos; Voelter, Markus; Noble, James (2006). Pattern Languages of Program Design 5. Addison-Wesley. ISBN 978-0-321-32194-7.



### Допоміжна

5. Douglass, Bruce (2004). Real-Time UML 3rd Edition. Newnes. ISBN 978-0321160768.
6. Douglass, Bruce (2002). Real-Time Design Patterns. Addison-Wesley Professional. ISBN 978-0201699562.
7. Douglass, Bruce (2009). Real-Time Agility. Addison-Wesley Professional. ISBN 978-0321545497.
8. Douglass, Bruce (2010). Design Patterns for Embedded Systems in C. Newnes. ISBN 978-1856177078.
9. Hohpe, Gregor; Woolf, Bobby (2003). Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley. ISBN 978-0-321-20068-6.
10. Freeman, Eric T.; Robson, Elisabeth; Bates, Bert; Sierra, Kathy (2004). Head First Design Patterns. O'Reilly Media. ISBN 978-0-596-00712-6.

### 12. Інформаційні ресурси

1. Занурення в ПАТЕРНИ ПРОЕКТУВАННЯ. Електронна книга про принципи та патерни проектування <https://refactoring.guru/uk/design-patterns/book>
2. <http://www.uml.org/>
3. <https://web.archive.org/web/20110906042707/http://www.itmweb.com/essay546.htm>
4. <https://web.archive.org/web/20120331162632/http://oss.org.cn/ossdocs/development/rup/webapps.htm>
5. <http://www.sereferences.com/uml-tools.php>
6. [http://blogs.oracle.com/JavaFundamentals/entry/the\\_importance\\_of\\_using\\_unified](http://blogs.oracle.com/JavaFundamentals/entry/the_importance_of_using_unified)