

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

Кафедра математичного моделювання та штучного інтелекту (№ 304)

**ЗАТВЕРДЖУЮ**

Гарант освітньої програми



(підпис)

Дмитро ЧУМАЧЕНКО

(ім'я та прізвище)

« 30 » 08 2024 р.

**РОБОЧА ПРОГРАМА *ОБОВ'ЯЗКОВОЇ* НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Проектування програмного забезпечення

(назва навчальної дисципліни)

Галузі знань: 12 "Інформаційні технології"

(шифр і найменування галузі знань)

Спеціальність: 122 "Комп'ютерні науки"

(код та найменування напрямку підготовки)

Освітня програма: "Інтелектуальні системи та технології"

(найменування освітньої програми)

**Форма навчання: денна**

**Рівень вищої освіти: перший (бакалаврський)**

**Харків 2024 рік**

Розробник: Базілевич К.О., доцент каф.304, к.т.н., доцент  
(прізвище та ініціали, посада, наукова ступінь та вчене звання)



(підпис)

Робочу програму навчальної дисципліни розглянуто на засіданні кафедри  
(№304) математичного моделювання та штучного інтелекту  
(назва кафедри)

Протокол № 1 від « 30 » серпня 2024 р.

Завідувач кафедри к.ф.-м.н., доц.  
(науковий ступінь та вчене звання)



О. В. Карташов  
(ініціали та прізвище)

## 1. Опис навчальної дисципліни

| Найменування показника   | Галузь знань, спеціальність, освітня програма, рівень вищої освіти   | Характеристика навчальної дисципліни (денна форма навчання) |
|--|--|---|
| Кількість кредитів – 6   | <b>Галузі знань</b><br><br><u>12 "Інформаційні технології"</u><br><small>(код і найменування)</small><br><br><b>Спеціальності</b><br><br><u>122 "Комп'ютерні науки"</u><br><small>(код і найменування)</small><br><br><b>Освітні програми</b><br><br><u>"Інтелектуальні системи та технології"</u><br><small>(найменування)</small><br><br><b>Рівень вищої освіти:</b><br><br>перший (бакалаврський) | <i>Обов'язкова</i>  |
| Кількість модулів – 3  |  | <b>Навчальний рік</b>                                       |
| Кількість змістовних модулів – 3   |  | 2024/2025   |
| Індивідуальне завдання – розрахункова робота та курсова робота (проект)                                  |  | <b>Семестр</b>  |
| Загальна кількість годин – 80/180  |  | 10-й  |
| Кількість тижневих годин для денної форми навчання: аудиторних – 2,5 самостійної роботи студента – 3,125 |  | <b>Лекції*</b>  |
|  |  | 32 години   |
|  |  | <b>Практичні, семінарські</b>                               |
|  |  | 16 годин  |
|  |  | <b>Лабораторні</b>  |
|  | 32 години  |   |
|  | <b>Самостійна робота</b>   |   |
| 100 годин  |  |   |
| <b>Вид контролю</b>  | модульний контроль, іспит  |   |

Співвідношення кількості годин аудиторних занять до самостійної і індивідуальної роботи становить:

для денної форми навчання – 80/100.

\*Аудиторне навантаження може бути зменшене або збільшене на одну годину залежно від розкладу занять.

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

**Мета:** забезпечення базової профілюючої підготовки за фахом, формування теоретичних знань та практичних навичок у галузі проектування інформаційних систем. Здобуті у процесі вивчення дисципліни знання є базою для вивчення дисциплін професійно-орієнтованого циклу.

**Завдання:** формування у студентів базових системних понять і навичок, цілісного бачення інформаційної системи, посилення міждисциплінарних зв'язків, розвиток системного мислення, без яких неможливе ефективне використання інформаційних технологій.

### **Формування компетентностей:**

ЗК2. Здатність застосовувати знання у практичних ситуаціях.

ЗК3. Знання та розуміння предметної області та професійної діяльності.

ЗК9 – Здатність працювати в команді.

СК3. Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв'язності та нерозв'язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.

СК8. Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.

### **Результати навчання:**

ПРН14. Застосовувати знання методології та CASE-засобів проектування складних систем, методів структурного аналізу систем, об'єктно-орієнтованої методології проектування при розробці і дослідженні функціональних моделей організаційно-економічних і виробничо-технічних систем.

**Пререквізити:** «Основи програмування», «Об'єктно-орієнтоване програмування».

**Кореквізити:** «Бази даних та інформаційні системи».

**Постреквізити:** «Розподілені інформаційно-аналітичні системи», «Кваліфікаційна робота».

### **3. Програма навчальної дисципліни**

#### **Модуль 1.**

#### **Змістовий модуль 1. Основи проектування програмного забезпечення інформаційних систем**

##### **Тема 1. Вступ до дисципліни**

*Поняття інформаційної системи (ІС). Класи ІС. Структура багатокористувальницької та однокористувальницької ІС, малої та корпоративної ІС, локальної і розподіленої ІС, склад і призначення підсистем. Основні особливості сучасних проектів ІС. Етапи створення ІС: формування вимог, концептуальне проектування, специфікація додатків, розробка моделей, інтеграція і тестування інформаційної системи. Методи програмної інженерії в проектуванні ІС.*

**Тема 2. Життєвий цикл програмного забезпечення ІС** *Поняття життєвого циклу ПО ІС. Процеси життєвого циклу: основні, допоміжні, організаційні. Зміст і взаємозв'язок процесів життєвого циклу ПО ІС. Моделі життєвого циклу: каскадна, модель з проміжним контролем, спіральна. Стадії життєвого циклу ПО ІС. Регламентація процесів проектування у вітчизняних і міжнародних стандартах. Канонічне проектування ІС. Стадії і етапи процесу канонічного проектування ІС. Цілі і завдання передпроектної стадії створення ІС. Моделі діяльності організації. Склад робіт на стадії технічного і робочого проектування. Склад проектної документації. Типове проектування ІС. Поняття типового проекту, передумови типізації. Об'єкти типізації. Методи типового проектування. Типове проектне рішення (ТПР). Класи і структура ТПР. Склад і зміст операцій типового елементного проектування ІС. Функціональні пакети прикладних програм (ППП) як основа ТПР. Адаптація типової ІС. Методи і засоби прототипного проектування ІС.*

**Тема 3. Аналіз і моделювання функціональної області впровадження ІС** *Основні поняття організаційного бізнес-моделювання. Дерево цілей і стратегії їх досягнення. Статичний і динамічний опис предметної області. Процесні потокові моделі. Моделі структур даних. Повна бізнес-модель. Шаблони організаційного бізнес-моделювання. Побудова організаційно-функціональної структури. Інформаційні технології організаційного моделювання. Основні елементи процесного підходу: межі процесу, ключові ролі, дерево цілей, дерево функцій, дерево показників. Виділення і класифікація процесів. Основні процеси, процеси управління, процеси забезпечення. Референтні моделі. Проведення передпроектного обстеження організації. Анкетування, інтерв'ювання, фотографія робочого часу персоналу. Результати передпроектного обстеження. Методології моделювання предметної області. Структурна модель предметної області. Об'єктна структура. Функціональна структура. Структура управління. Організаційна структура. Функціонально-орієнтовані і об'єктно-орієнтовані методології опису предметної області. Функціональна методика IDEF. Функціональна*

методика потоків даних. Об'єктно-орієнтована методика. Порівняння існуючих методик. Синтетична методика. Case-засоби для моделювання ділових процесів. Інструментальні середовища. Принципи побудови моделі IDEF0: контекстна діаграма, суб'єкт моделювання, мета і точка зору. Діаграми IDEF0: контекстна діаграма, діаграми декомпозиції, діаграми дерева вузлів, діаграми тільки для експозиції (FEO). Вартісний аналіз: об'єкт витрат, двигун витрат, центр витрат. Властивості, визначувані користувачем (UDP). Діаграми потоків даних (Data Flow Diagramming): роботи, зовнішня суть (посилання), потоки робіт, сховища даних. Метод опису процесів IDEF3: роботи, зв'язки, об'єкти посилань, перехрестя. Імітаційне моделювання: джерела і стоки, черги, процеси.

**Тема 4. Основні поняття менеджменту розробки програмних виробів.** Функціональні ролі в колективі розробників, модель MSF. Планування і контроль розвитку проекту. Цикл управління проектом. Результативність проектної діяльності програміста. Управління ризиками і якістю.

#### **Тема 5. Сучасні методології розробки програмного забезпечення**

Введення в раціональний процес моделювання RUP. Основи процесу розробки ІС відповідно до методології RUP. Загальне уявлення про процес. Початкова фаза проекту. Уточнення. Ризики, пов'язані з вимогами. Технологічні ризики. Ризики, пов'язані з кваліфікацією персоналу. Політичні ризики. Базова архітектура. Конструювання. Введення в дію. Артефакти. Робочі процеси. Agile-технології: SCRUM, Extremal Programming, Kanban

#### **Модульний контроль.**

#### **Змістовий модуль 2. Проектування ІС із застосуванням UML**

**Тема 6. Етапи проектування ІС із застосуванням UML** Діаграми в UML. Класи і стереотипи класів. Асоціативні класи. Основні елементи діаграм взаємодії – об'єкти, повідомлення. Діаграми станів: початкового стану, кінцевого стану, переходи. Вкладеність станів. Діаграми впровадження: підсистеми, компоненти, зв'язки. Стереотипи компонент. Діаграми розміщення. Основні типи UML-діаграм, використовувани в проектуванні інформаційних систем. Взаємозв'язки між діаграмами. Підтримка UML ітеративного процесу проектування ІС: моделювання бізнес-прецедентів, розробка моделі бізнес-об'єктів, розробка концептуальної моделі даних, розробка вимог до системи, аналіз вимог і попереднє проектування системи, розробка моделей даних і додатків, проектування фізичної реалізації системи.

#### **Модульний контроль.**

#### **Змістовий модуль 3. Тестування програмного забезпечення**

**Тема 7. Основні поняття тестування** Якість. Значення якості для підприємства. Процес (стадії) розробки ПО з схемою. Роль QA на всіх етапах проекту. Quality assurance vs. testing – відмінність понять. Поняття тестування ПО. Його принципи. Основні терміни тестування. Підходи до обґрунтування істинності формул і програм і їх зв'язок з тестуванням. Основні проблеми тестування, завдання вибору кінцевого набору тестів.

**Тема 8. Методології тестування. Процес тестування. Аксиоми тестування. Статичне і динамічне тестування. Автоматизоване і ручне тестування. Альфа і бета-тестування. Тест-кейс і тест-світ. Критерії вибору тестів. Оцінка відтестованості проекту: метрики і методика інтегральної оцінки. Типи тестування: функціональне тестування (Functionality testing), навантажувальне тестування (Stress testing), конфігураційне тестування (Configuration testing), інсталяційне тестування (Installation testing), метод "Білого ящика" (White-box testing), метод "Чорного ящика" (Black-box testing), метод "Сірого ящика" (Gray-box testing), тестування ергономіки (Usability testing), тестування наростаючої інтеграції (Incremental integration testing), поверхнева перевірка (Sanity Test or "Smoke Test").**

**Тема 9. Документування і оцінка тестування. Автоматизація тестування. Структура тестового набору для автоматичного прогону. Структура інструментальної системи автоматизації тестування. Особливості індустріального тестування. Структура документа "Тестовий план". Побудування матриці покриття тестами. Типи тестування для різних частин продукту або для перевірки різних характеристик продукту. Підходи до тестування специфікацій і сценаріїв. Ручний підхід і підхід генерації тестових наборів при розробці тестів. Особливості документування тестових процедур для ручних і автоматизованих тестів.**

#### **Модульний контроль.**

#### **4. Структура навчальної дисципліни**

| Назви змістових модулів і тем  | Кількість годин |              |     |      |    |
|--|-----------------|--------------|-----|------|----|
|  | денна форма     |              |     |      |    |
|  | усього          | у тому числі |     |      |    |
| л  |                 | п            | лаб | с.р. |    |
| 1  | 2               | 3            | 4   | 5    | 6  |
| <b>Модуль 1</b>  |                 |              |     |      |    |
| <b>Змістовий модуль 1. Основи проектування програмного забезпечення інформаційних систем</b> |                 |              |     |      |    |
| Тема 1. Вступ до дисципліни  | 4               | 2            | –   | –    | 2  |
| Тема 2. Життєвий цикл програмного забезпечення ІС  | 4               | 2            | –   | –    | 2  |
| Тема 3. Аналіз і моделювання функціональної області впровадження ІС.                         | 10              | 2            | –   | 4    | 4  |
| Тема 4. Основні поняття менеджменту розробки програмних виробів                              | 12              | 2            | –   | 4    | 6  |
| Тема 5. Сучасні методології розробки програмного забезпечення                                | 8               | 4            | –   | –    | 4  |
| Модульний контроль   | 2               | 2            | –   | –    | –  |
| Разом за змістовим модулем 1   | 40              | 14           | –   | 8    | 18 |

| 1  | 2   | 3  | 4  | 5  | 6   |
|--|-----|----|----|----|-----|
| <b>Змістовий модуль 2. Проектування ІС із застосуванням UML</b>              |     |    |    |    |     |
| Тема 6. <i>Етапи проектування ІС із застосуванням UML</i>                    | 50  | 12 | –  | 20 | 18  |
| Модульний контроль   | 2   | 2  | –  | –  | –   |
| Разом за змістовим модулем 2   | 52  | 14 | –  | 20 | 18  |
| <b>Змістовий модуль 3. Тестування програмного забезпечення</b>               |     |    |    |    |     |
| Тема 7. <i>Основні поняття тестування</i>                                    | 3   | 1  | –  |    | 2   |
| Тема 8. <i>Методології тестування.</i>                                       | 5   | 1  | –  |    | 4   |
| Тема 9. <i>Документування і оцінка тестування. Автоматизація тестування.</i> | 8   | –  | –  | 4  | 4   |
| Модульний контроль   | 2   | 2  | –  | –  | –   |
| Разом за змістовим модулем 3   | 18  | 4  | –  | 4  | 10  |
| <b>Модуль 2.</b>   |     |    |    |    |     |
| Індивідуальне завдання – розрахункова робота                                 | 10  | –  | –  | –  | 10  |
| <b>Модуль 3.</b>   |     |    |    |    |     |
| Індивідуальне завдання – курсова робота (проект)                             | 60  | –  | 16 | –  | 44  |
| <b>Контрольний захід.</b>  |     |    |    |    |     |
| <b>Усього годин</b>  | 180 | 32 | 16 | 32 | 100 |

### 5. Теми семінарських занять

| № з/п | Назва теми   | Кількість годин |
|-------|--------------|-----------------|
| 1     |              |                 |
| 2     |              |                 |
|       | <b>Разом</b> |                 |

### 6. Теми практичних занять

| № з/п | Назва теми                     | Кількість годин |
|-------|--------------------------------|-----------------|
| 1     | Консультації з курсової роботи | 16              |
|       | <b>Разом</b>                   | 16              |



## 7. Теми лабораторних занять

| № з/п | Назва теми  | Кількість годин |
|-------|---|-----------------|
| 1     | Побудова функціональної моделі системи за допомогою методології IDEF0. Моделювання бізнес процесів з використанням IDEF3. | 4               |
| 2     | Застосування RUP для формування бачення проекту.  | 4               |
| 3     | Вивчення UML. Розробка діаграм варіантів використання.  | 4               |
| 4     | Вивчення UML. Розробка діаграм взаємодії.   | 4               |
| 5     | Вивчення UML. Розробка діаграм класів.  | 4               |
| 6     | Вивчення UML. Розробка діаграм стану та діяльності  | 4               |
| 7     | Вивчення UML. Розробка діаграм компонентів системи і діаграми розміщення. Генерація коду.                                 | 4               |
| 8     | Автоматизація тестування на базі Selenium IDE   | 4               |
|       | <b>Разом</b>  | <b>32</b>       |

## 8. Самостійна робота

| № з/п | Назва теми   | Кількість годин |
|-------|--|-----------------|
| 1     | Побудова функціональної моделі системи за допомогою методології SADT                                   | 2               |
| 2     | Альтернативні UML-оболонки: Enterprise Architect   | 2               |
| 3     | Альтернативні UML-оболонки: Umbrello   | 4               |
| 4     | СММ-модель стиглості   | 6               |
| 5     | Вивчення моделей й методів тестування та оцінки якості програмного забезпечення                        | 4               |
| 6     | Життєвий цикл дефектів ПЗ  | 34              |
| 7     | Вивчення моделей обчислення результативності роботи програміста. Побудування календарного плану робіт. | 2               |
| 8     | Bug-трекінгова система Bugzilla  | 4               |
| 9     | Шаблони проектування   | 4               |
| 10    | Виконання розрахункової роботи   | 10              |
| 11    | Виконання курсової роботи (проекта)  | 28              |
|       | <b>Разом</b>   | <b>100</b>      |

## 9. Індивідуальні завдання

Розрахункова робота на тему «Паттерни проектування та їх використання».

Курсова робота (проект) «Розробка прикладного програмного забезпечення з використанням сучасних засобів проектування».

## 10. Методи навчання

1. Пояснювально-ілюстративний (інформаційно-рецептивний) метод: та метод проблемного виконання (лекція).
2. Репродуктивний (лабораторні роботи).
3. Частково-пошуковий (евристичний) та дослідницький: (самостійна робота та виконання РР).

## 11. Методи контролю

Визначення рівня засвоєння студентом навчального матеріалу дисципліни здійснюється шляхом проведення поточних і підсумкових контролів. У завдання поточного контролю входить систематична перевірка розуміння й засвоєння студентом програмного матеріалу, виконання лабораторних робіт, уміння самостійно проробляти тексти складання конспектів, здатності усно або письмово представляти певний матеріал. Перед підсумковим контролем ставиться завдання перевірки глибини засвоєння студентом програмного матеріалу дисципліни, логіки й взаємозв'язки між її окремими розділами, здатності творчо використати придбані знання, уміння сформулювати своє відношення до проблеми, що впливає зі змісту дисципліни.

## 12. Розподіл балів, які отримують студенти

12.1. Розподіл балів, які отримують студенти (кількісні критерії оцінювання для одержання позитивної оцінки)

| Складові навчальної роботи              | Бали за одне заняття (завдання) | Кількість занять (завдань) | Сумарна кількість балів |
|---|---------------------------------|----------------------------|-------------------------|
| <b>Змістовний модуль 1</b>              |                                 |                            |                         |
| Виконання і захист лабораторних робіт   | 3...5                           | 8                          | 24...40                 |
| Модульний контроль                      | 10...30                         | 1                          | 10...30                 |
| <b>Змістовний модуль 2</b>              |                                 |                            |                         |
| Виконання і захист розрахункової роботи | 26...30                         | 1                          | 26...30                 |
| <b>Усього за семестр</b>                |                                 |                            | <b>60...100</b>         |

| Змістовний модуль 3                |          |   |                |
|------------------------------------|----------|---|----------------|
| Виконання і захист курсової роботи | 60...100 | 1 | 60...100       |
| <b>Усього за семестр</b>           |          |   | <b>0...100</b> |

## 12.2. Якісні критерії оцінювання

Необхідний обсяг знань для одержання позитивної оцінки:

- основні особливості сучасних проектів інформаційних систем;
- етапи створення програмного забезпечення інформаційних систем;
- моделі життєвого циклу програмного продукту;
- основні поняття організаційного бізнес-моделювання;
- методології моделювання предметної області;
- основи процесу розробки ІС відповідно до сучасних методологій (RUP, SCRUM, MSF);
- основні поняття та методології тестування програмного забезпечення.

Необхідний обсяг вмінь для одержання позитивної оцінки (у тому числі використовуючи готові програмні рішення та спеціалізовані програмні бібліотеки):

- створювати функціональні моделі систем за допомогою методології IDEF0;
- здійснювати моделювання бізнес процесів з використанням IDEF3;
- супроводжувати процес побудування програмного забезпечення за допомогою розробки UML;
- проводити обчислення результативності роботи програміста;
- застосовувати RUP для формування бачення проекту
- розробляти план тестування та складати протоколи тестування програмного забезпечення.

## 12.3 Критерії оцінювання роботи студента протягом семестру

**Задовільно (60-74).** Мати мінімум знань та умінь. Відпрацювати та захистити всі лабораторні роботи. Розуміти опис: функціональної моделі системи у вигляді діаграми IDEF0, бізнес процесів у вигляді діаграми IDEF3, програмного забезпечення за діаграм UML.

**Добре (75 - 89).** Твердо знати мінімум знань, виконати усі завдання. Показати вміння виконувати та захищати всі лабораторні роботи в обумовлений викладачем строк з обґрунтуванням рішень та заходів, які запропоновано у роботах. Розуміти та доповнювати опис: функціональної моделі системи у вигляді діаграми IDEF0, бізнес процесів у вигляді діаграми IDEF3, програмного забезпечення за діаграм UML. Вміти створювати план тестування та складати протоколи тестування програмного забезпечення.

**Відмінно (90 - 100).** Повно знати основний та додатковий матеріал. Знати усі теми. Орієнтуватися у підручниках та посібниках. Вміти самостійно створювати функціональні моделі систем за допомогою методології IDEF0, здійснювати моделювання бізнес процесів з використанням IDEF3, супроводжувати процес побудування програмного забезпечення за допомогою розробки UML, проводити обчислення результативності роботи програміста,

розробляти план тестування та складати протоколи тестування програмного забезпечення. Застосовувати на практиці методології та паттерни для розробки програмного забезпечення

Розподіл балів, які отримують студенти за виконання курсової роботи

|                      |                      |               |      |
|----------------------|----------------------|---------------|------|
| Пояснювальна записка | Ілюстративна частина | Захист роботи | Сума |
| до 60 балів          | до 15 балів          | до 25 балів   | 100  |

### 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 |

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

### 13. Методичне забезпечення

Методичне забезпечення:

- робоча програма дисципліни;
- конспект лекцій, підручники (навчальні посібники);
- методичні вказівки та рекомендації для виконання курсових робіт та проєктів, розрахункових та розрахунково-графічних робіт, лабораторних та практичних робіт, а також рекомендації для самостійної підготовки;
- тематики індивідуальних завдань;
- приклади розв'язування типових задач чи виконання типових завдань;
- питання, тести для контрольних заходів.

Посилання курс у системі дистанційного навчання Ментор

<https://mentor.khai.edu/course/view.php?id=5043>

### 14. Рекомендована література

#### Базова

1. Канер Сем. Тестування програмного забезпечення. Фундаментальні концепції менеджменту бізнес-додатків: Пер. с англ. – К. : Видавництво «ДіаСофт», 2001. – 544 с.
2. С.В.Баран. Розробка програмного забезпечення з використанням патернів проєктування: Навчальний посібник. – Кривий Ріг: Державний університет економіки і технологій, 2023. –203 с.

3. Paul Clements; Felix Bachmann; Len Bass; David Garlan; James Ivers; Reed Little; Paulo Merson; Robert Nord; Judith Stafford. Documenting Software Architectures: Views and Beyond. – Addison-Wesley Professional, 2010.

4. Len Bass, Paul Clements, Rick Kazman: Software Architecture in Practice, Third Edition. Addison Wesley, 2012.

#### **Допоміжна**

1. Grand M. Patterns in Java. A Catalog of Reusable Design Patterns Illustrated with UML – Wiley & Sons, 1998. Vol. 1, - 468 p. Vol. 2, – 356 p.

2. Starr L. Executable UML. How to build class models - Prentice Hall PTR, 2002. – 418 p.

3. Wampler B.E. The Essence Object-Oriented Programming with Java and UML - Addison-Wesley, 2002, – 290 p.

### **15. Інформаційні ресурси**

<http://k304.khai.edu/> - сайт кафедри 304