

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ХАРКІВСЬКИЙ НАЦІОНАЛЬНИЙ АВТОМОБІЛЬНО-ДОРОЖНІЙ
УНІВЕРСИТЕТ**

Кафедра комп'ютерних технологій і мехатроніки

ЗАТВЕРДЖЕНО

Гарант освітньо-професійної програми
«Інженерія програмного забезпечення»
першого(бакалаврського) рівня освіти:
зав. каф. КТМ, д.т.н., проф.
 Ніконов О.Я.

**СИЛАБУС
АРХІТЕКТУРА ТА АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО
ЗАБЕЗПЕЧЕННЯ /
ARCHITECTURE AND ANALYSIS OF SOFTWARE
REQUIREMENTS
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 2020

Автор: Шапошнікова Олена Павлівна, доцент кафедри комп'ютерних технологій і мехатроніки

Силабус розглянуто та затверджено на засіданні кафедри комп'ютерних технологій і мехатроніки, протокол № 18 від «27» червня 2020 р.

СИЛАБУС

АРХІТЕКТУРА ТА АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ /

ARCHITECTURE AND ANALYSIS OF SOFTWARE REQUIREMENTS

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

Анотація курсу

1. Викладачі

1.1. Лектор: Шапошнікова Олена Павлівна

- доцент кафедри комп'ютерних технологій та мехатроніки;
- педагогічний стаж – 25 років
- контактний телефон +38-057-707-37-43
- e-mail: shaposhnikovaep@gmail.com
- наукові інтереси: архітектура ПЗ, аналіз вимог до ПЗ, управління ІТ проектами, якість ПЗ та тестування.

1.2. Асистент лектора:

2. Дисципліна «Архітектура та аналіз вимог до програмного забезпечення»

- рік навчання: 2;
- семестр навчання: 4;
- кількість годин за семестр: 120, в т.ч.
лекційних: 16;
практичних занять: 32;
на самостійне опрацювання: 72;
- кількість аудиторних годин на тиждень
лекційних: 2 (раз на два тижні);
практичних занять: 2.

3. Час та місце проведення

- аудиторні заняття – відповідно до розкладу ХНАДУ, ауд. 214, 313;
- позааудиторна робота – самостійна робота студента із використанням технологій віртуалізації MS Visio, Visual Use case.

4. Пререквізити та постреквізити навчальної дисципліни:

- **перереквізити:** «Основи інформаційних технологій», «Алгоритмізація та програмування», «Операційні системи», «Об'єктно-орієнтоване програмування», «Алгоритми і структури даних».
- **постреквізити:** «Програмування баз даних», «Web-програмування», «Управління ІТ проектами», «Якість програмного забезпечення та тестування».1

5. Характеристика дисципліни:

5.1. Призначення навчальної дисципліни: вивчення дисципліни «Архітектура та аналіз вимог до програмного забезпечення» займає вагомe місце у структурі отримання знань за освітньою програмою «Програмне забезпечення систем». Процес проектування програмного забезпечення (ПЗ) починається із вивчення предметної області, формулювання концепції проекту, визначення його границь, виявлення зацікавлених осіб та формулювання вимог до ПЗ. Наступним етапом є процес розроблення архітектури, який дозволяє знизити складність системи шляхом абстракції та розмежування повноважень. Архітектура ПЗ об'єднує різні точки зору на систему, що є вагомим аргументом щодо до необхідності та доцільності розроблення архітектури ПЗ. Ця дисципліна формує системний підхід до проектування ПЗ та інженерний світогляд фахівця у галузі ІТ.

5.2. Мета вивчення дисципліни: «Архітектура та аналіз вимог до програмного забезпечення» є теоретична та практична підготовка студентів, яка

має забезпечити отримання ними основних знань у галузі сучасних технологій проектування, інженерії вимог до програмного забезпечення (ПЗ) та вивчення архітектури програмного забезпечення, отримання практичних навичок реалізації програмних систем, основи моделювання і аналізу програмних систем, аналізу розробки, специфікації та управління вимогами.

5.3. Задачі вивчення дисципліни: є формування сукупності знань та вмінь для застосування на практиці уніфікованого процесу проектування та розробки програмних систем, об'єктно-орієнтованого аналізу і проектування, документування архітектури програмних систем, застосовування основних архітектурних стилів і шаблонів проектування програмного забезпечення. Професійні компетентності, які отримують студенти після вивчення навчальної дисципліни:

Інтегральна компетентність:

здатність використовувати поглиблені теоретичні та фундаментальні знання, уміння і навички для успішного розв'язування спеціалізованих та практичних задач під час професійної діяльності у галузі інформаційних технологій.

Загальні компетентності:

- здатність до абстрактного мислення, аналізу та синтезу;
- здатність застосовувати знання у практичних ситуаціях;
- здатність спілкуватися іноземною мовою як усно, так і письмово;
- здатність до пошуку, оброблення та аналізу інформації з різних джерел;
- здатність роботи в команді;
- здатність вчитися і оволодівати сучасними знаннями.

Спеціальні (фахові, предметні) компетентності:

- здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення;
- здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування;
- здатність розробляти архітектури, модулі та компоненти програмних систем;
- здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення;
- здатність до алгоритмічного та логічного мислення.

Програмні результати навчання:

- знати, аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки;
- знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення;
- знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізів та математичного моделювання для розробки програмного забезпечення;

- знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення;
- проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування;
- вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання;
- знати ефективні підходи щодо проектування 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)
1	2	3	4
РОЗДІЛ 1. АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ			
Тема 1. Вступ до проблеми розробки вимог		14	11
Загальні та спеціальні компетентності: здатність застосовувати знання у практичних ситуаціях, до пошуку, оброблення та аналізу інформації з різних джерел; здатність спілкуватися іноземною мовою як усно, так і письмово; здатність роботи в команді; знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення. Результати навчання: формулювати та аналізувати вимоги до програмного забезпечення; вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання;	Лекція №1. Вступ до проблеми розробки вимог <i>План лекції:</i> 1. Контекст управління вимогами до ПЗ. 2. Роль аналітика. 3. Вимоги. 4. Процес розробки вимог. Список рекомендованих джерел: Основний: 1-5 Додатковий: 1-3 Інтернет-ресурси: 1-3	2	
	Завдання для самостійної роботи: Вивчення та доповнення матеріалів лекції на основі самостійного опрацювання основних літературних джерел, зазначених у списку на електронних матеріалів, які надаються викладачем Питання, внесені на самостійне опрацювання: 1. Поняття про предметну область проекту. 2. Формальні визначення. 3. Класифікація моделей.	8	
	Практичне заняття № 1. «Аналіз предметної області. Визначення концепції проекту» Мета: Отримати навички виявлення реальних потреб замовника та з'ясування сенсу висловлених вимог, а також розроблення концепції проекту. Завдання: 1. Ознайомитися з видами моделей. 2. Ознайомитися із методологіями IDEF. 3. Вивчити синтаксис IDEF0-моделей. План заняття:	4	

	–Актуалізація теоретичного матеріалу –Виконання завдань практичної роботи –Презентація виконаної роботи		
Тема 2. Виявлення вимог та потреб		14	13
<i>Загальні та спеціальні компетентності:</i> здатність застосовувати знання у практичних ситуаціях, до пошуку, оброблення та аналізу інформації з різних джерел, здатність роботи в команді; здатність ідентифікувати, класифікувати та формулювати вимоги до програмного забезпечення; знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення; <i>Результати навчання:</i> формулювати та аналізувати вимоги до програмного забезпечення; вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання; проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування;	Лекція 2. Виявлення вимог та потреб <i>План лекції:</i> 1. Робота з зацікавленими особами. 2. Збір первинних вимог. 3. Техніки для виявлення вимог та потреб. 4. Класифікація вимог. Список рекомендованих джерел: Основний: 1-7, 9 Додатковий: 1-2 Інтернет-ресурси: 1-4	2	
	Завдання для самостійної роботи: Вивчення та доповнення матеріалів лекції на основі самостійного опрацювання основних літературних джерел, зазначених у списку на електронних матеріалів, які надаються викладачем Питання, внесені на самостійне опрацювання: Case-засоби для розроблення вимог до ПЗ. 1. Поняття про CASE-засоби для розроблення вимог; 2. Необхідні та достатні функції CASE-засобу для розроблення вимог; 3. Виконати аналітичний огляд CASE-засобів;	8	
	Практичне заняття № 2. Побудова функціональної моделі проекту та її декомпозиція. <i>Мета:</i> навчитися створювати модель системи в стандарті IDEF0. <i>Завдання:</i> 1. Розробити контекстну діаграму. 2. Створити діаграму декомпозиції. 3. Створити діаграму дерева вузлів. <i>План заняття:</i> –Актуалізація теоретичного матеріалу –Виконання завдань практичної роботи –Презентація виконаної роботи	4	
Тема 3. Визначення системи		14	13
<i>Загальні та спеціальні компетентності:</i> здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення; здатність до алгоритмічного та логічного мислення,	Лекція 3. Визначення системи <i>План лекції:</i> 1. Концепція системи. 2. Представлення зацікавлених осіб. 3. Границі системи/продукту. Модель сценаріїв використання. Список рекомендованих джерел: Основний: 2, 4, 5 Додатковий: 2, 8 Інтернет-ресурси: 1-4	2	
	Завдання для самостійної роботи:	8	

знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення; здатність спілкуватися іноземною мовою як усно, так і письмово. <i>Результати навчання:</i> знати, аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки; знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення; знати та вміти використовувати методи та засоби збору, формулювання та аналізу вимог до програмного забезпечення; проводити передпроектне обстеження предметної області, системний аналіз об'єкта проектування; вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання;	Вивчення та доповнення матеріалів лекції на основі самостійного опрацювання основних літературних джерел, зазначених у списку на електронних матеріалів, які надаються викладачем Питання, внесені на самостійне опрацювання: Методологія BPMN. 1. Поняття методології BPMN; 2. Синтаксис BPMN; 3. Ознаки класифікації подій.		
	Практичне заняття № 3. Розроблення дерева функцій проекту та вимог до проекту. <i>Мета:</i> навчитися аналізувати і синтезувати систему за допомогою дерева функцій <i>Завдання:</i> 1. Визначити цільову функцію; 2. Визначити основні функції; 3. Визначити сервісні функції; 4. Сформувати дерево функцій. <i>План заняття:</i> –Актуалізація теоретичного матеріалу –Виконання завдань практичної роботи –Презентація виконаної роботи	4	
РОЗДІЛ 2. АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ			
Тема 4. Основи побудови архітектури ІС		14	11
<i>Загальні та спеціальні компетентності:</i> здатність до абстрактного мислення, аналізу та синтезу; здатність застосовувати знання у практичних ситуаціях; здатність до пошуку, оброблення та аналізу інформації з різних джерел;	Лекція 4. Тема 4. Основи побудови архітектури ІС. <i>План лекції:</i> 1. Поняття архітектури ПЗ. 2. Класифікація архітектур. 3. Аналітичний огляд архітектур Список рекомендованих джерел: Основний: 4, 5, 7, 14 Додатковий: 4, 5 Інтернет-ресурси: 8, 9	2	
здатність роботи в команді;	Завдання для самостійної роботи: Вивчення та доповнення матеріалів лекції на основі самостійного опрацювання	8	

здатність вчитися і оволодівати сучасними знаннями. здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування; здатність розробляти архітектури, модулі та компоненти програмних систем; <i>Результати навчання:</i> знати, аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки; знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення;	основних літературних джерел, зазначених у списку на електронних матеріалів, які надаються викладачем Питання, внесені на самостійне опрацювання: Еволюція платформних архітектур. 1. Автономна архітектура, 2. Централізована архітектура 3. Розподільна архітектура: – Архітектура «файл-сервер», – Архітектура «клієнт-сервер», – Архітектура веб-додатків.		
	Практичне заняття № 4. Моделювання поведінки системи на логічному рівні (розроблення діаграми станів) <i>Мета: навчитися аналізувати поведінку об'єкта в декількох прецедентах.</i> <i>Завдання:</i> 1. Визначити клас проекту для якого буде розроблена діаграма станів, 2. Вивчити основний набір символів UML для створення діаграми. 3. Розробити діаграму. 4. Проаналізувати стани та способи реакції на події. <i>План заняття:</i> –Актуалізація теоретичного матеріалу –Виконання завдань практичної роботи –Презентація виконаної роботи	4	
Тема 5. Архітектури ПЗ		16	12
<i>Загальні та спеціальні компетентності:</i> здатність до абстрактного мислення, аналізу та синтезу; здатність застосовувати знання у практичних ситуаціях; здатність роботи в команді; здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування; здатність розробляти архітектури, модулі та компоненти програмних	Лекція 5. Архітектури ПЗ <i>План лекції:</i> 1. Вибір структурних елементів та їх інтерфейсів. 2. З'єднання елементів структури. 3. Архітектурний стиль. Список рекомендованих джерел: Основний: 7-11 Додатковий: 4-7 Інтернет-ресурси: 10, 11	2	
здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування; здатність розробляти архітектури, модулі та компоненти програмних	Завдання для самостійної роботи: Вивчення та доповнення матеріалів лекції на основі самостійного опрацювання основних літературних джерел, зазначених у списку на електронних матеріалів, які надаються викладачем Питання, внесені на самостійне опрацювання: Розподілені архітектури. 1. Класифікація програмних систем. 2. Основні ознаки розподілених систем.	10	

систем; здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань інженерії програмного забезпечення; здатність до алгоритмічного та логічного мислення. <i>Результати навчання:</i> знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення; знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізів та математичного моделювання для розробки програмного забезпечення.	3. Недоліки розподілених систем. 4. Проблеми проектування розподілених систем. Практичне заняття № 5. Моделювання поведінки програмної системи на логічному рівні (розроблення діаграми послідовності). <i>Мета:</i> навчитися моделювати взаємодію об'єктів системи в часі, а також обмін повідомленнями між ними. <i>Завдання:</i> 1. Вивчити основний набір символів для створення діаграми. 2. Визначити поняття Об'єкт і Повідомлення. 3. Розробити діаграму. 4. Проаналізувати отриману діаграму. <i>План заняття:</i> –Актуалізація теоретичного матеріалу –Виконання завдань практичної роботи –Презентація виконаної роботи	4	
Тема 6. Багатошарова архітектура ПЗ		16	14
<i>Загальні та спеціальні компетентності:</i> здатність до абстрактного мислення, аналізу та синтезу; здатність застосовувати знання у практичних ситуаціях; здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування; здатність розробляти архітектури, модулі та компоненти програмних систем; здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв'язання завдань	Лекція 6. Багатошарова архітектура ПС <i>План лекції:</i> 1. Логічна структура багатошарової архітектури. 2. Поняття рівнів та слоїв. 3. Шаблони проектування багатошарової архітектури. 4. Проектування слоїв. Список рекомендованих джерел: Основний: 7-14, 16 Додатковий: 4, 5, 7 Інтернет-ресурси: 7, 12, 13	2	
	Завдання для самостійної роботи: Вивчення та доповнення матеріалів лекції на основі самостійного опрацювання основних літературних джерел, зазначених у списку на електронних матеріалів, які надаються викладачем Питання, внесені на самостійне опрацювання: Архітектура Web-додатків 1. Складові частини Web-додатку. 2. Архітектура багаторівневого Web-додатку. 3. Архітектура змішаного Web-додатку. 4. Архітектура багаторівневого Web-додатку на основі технології CORBA.	10	

інженерії програмного забезпечення; <i>Результати навчання:</i> знати, аналізувати, цілеспрямовано шукати і вибирати необхідні для вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки; знати і застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізів та математичного моделювання для розробки програмного забезпечення; знати ефективні підходи щодо проектування програмного забезпечення.	5. Архітектура Web-додатків з інтерфейсами OLE DB, ADO и ODBC. Практичне заняття № 6. Моделювання поведінки ІС на логічному рівні (розроблення діаграми кооперації) <i>Мета:</i> навчитися створювати специфікації об'єктів окремих класів, які взаємодіють з метою реалізації окремих варіантів використання в загальному контексті ПС шляхом розроблення діаграми кооперації. <i>Завдання:</i> 1. Вивчити основний набір символів для створення діаграми. 2. Визначити структурні елементи діаграми кооперації. 3. Розробити діаграму. 4. Проаналізувати отриману діаграму. <i>План заняття:</i> –Актуалізація теоретичного матеріалу –Виконання завдань практичної роботи –Презентація виконаної роботи	4	
Тема 7. Паттерни проектування		20	13
<i>Загальні та спеціальні компетентності:</i> здатність до абстрактного мислення, аналізу та синтезу; здатність до пошуку, оброблення та аналізу інформації з різних джерел; здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування; здатність розробляти архітектури, модулі та компоненти програмних систем; здатність до алгоритмічного та логічного мислення. <i>Результати навчання:</i> знати, аналізувати, цілеспрямовано шукати і вибирати необхідні для	Лекція 7. Паттерни проектування <i>План лекції:</i> 1. Поняття паттерна проектування. 2. Види паттернів. 3. Породжуючі шаблони. 4. Структурні шаблони. 5. Поведінкові шаблони. Список рекомендованих джерел: Основний: 7-13, 16 Додатковий: 4-7 Інтернет-ресурси: 14-17	2	
	Завдання для самостійної роботи: Вивчення та доповнення матеріалів лекції на основі самостійного опрацювання основних літературних джерел, зазначених у списку на електронних матеріалів, які надаються викладачем Питання, внесені на самостійне опрацювання: Сокети. 1. Поняття сокету. 2. Принципи сокетів. 3. Функції сокетів.	10	
	Практичне заняття № 7. Розроблення моделі фізичного представлення ІС (розроблення діаграми компонентів та розгортання)	8	

вирішення професійних завдань інформаційно-довідникові ресурси і знання з урахуванням сучасних досягнень науки і техніки; знати ефективні підходи щодо проектування програмного забезпечення.	<i>Мета: навчитися розділяти систему на компоненти та представляти загальну конфігурацію і топологію ПС.</i> <i>Завдання:</i> 1. Вивчити основний набір символів для створення діаграм. 2. Визначити з яких фізичних часток складається ПС. 3. Визначити типи користувачів. 4. Визначити апаратні та технічні типи пристроїв для виконання системою своїх функцій. 5. Визначити види та потрібну пропускну здатність каналів зв'язку. 6. Розробити діаграми. <i>План заняття:</i> –Актуалізація теоретичного матеріалу –Виконання завдань практичної роботи –Презентація виконаної роботи		
Тема 8. Прийоми об'єктно-орієнтованого проектування.		12	13
<i>Загальні та спеціальні компетентності:</i> здатність до абстрактного мислення, аналізу та синтезу; здатність застосовувати знання у практичних ситуаціях; здатність роботи в команді; здатність вчитися і оволодівати сучасними знаннями. здатність брати участь у проектуванні програмного забезпечення, включаючи проведення моделювання (формальний опис) його структури, поведінки та процесів функціонування; здатність розробляти архітектури, модулі та компоненти програмних систем; здатність до алгоритмічного та логічного мислення. <i>Результати навчання:</i> знати основні процеси, фази та ітерації життєвого циклу програмного забезпечення; знати і	Лекція 8. Прийоми об'єктно-орієнтованого проектування <i>План лекції:</i> 1. Головні принципи об'єктно-орієнтованої архітектури (OOA). 2. Переваги OOA. 3. Концепції та елементи DDD. Список рекомендованих джерел: Основний: 7-14, 16 Додатковий: 4-7 Інтернет-ресурси: 18 Завдання для самостійної роботи: Вивчення та доповнення матеріалів лекції на основі самостійного опрацювання основних літературних джерел, зазначених у списку на електронних матеріалів, які надаються викладачем Питання, внесені на самостійне опрацювання: СОМ-технологія 1. Поняття СОМ-технології. 2. Проблеми, які вирішуються за допомогою СОМ-технології.	2 10	

застосовувати відповідні математичні поняття, методи доменного, системного і об'єктно-орієнтованого аналізів та математичного моделювання для розробки програмного забезпечення; вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання;			
Разом		120 годин/ 4 кредити	100 балів
Підсумковий контроль		Залік	

Рекомендовані джерела інформації

1. Базова література

1. Вигерс Карл, Битти Джой Разработка требований к программному обеспечению. 3-е изд., дополненное / Пер. с англ. — М.: Издательство «Русская редакция»; СПб.: БХВ-Петербург, 2014. — 736 стр.: ил.
2. Вендров, А.М. CASE-технологии. Современные методы и средства проектирования информационных систем / А.М. Вендров. — М.: Финансы и статистика, 1998. — 176 с.
3. Марка, Д.А. Методология структурного анализа и проектирования SADT / Д.А. Марка, К. МакГоуэн. — М.: МетаТехнология, 1993. — 243 с.
4. Калянов, Г.Н. CASE. Структурный системный анализ (автоматизация и применение) / Г.Н. Калянов. — М.: Лори, 1996. — с.
5. Баркер, Р. CASE*Method. Моделирование взаимосвязей между сущностями / Р. Баркер. — М., 1992. — 233 с.
6. Маклаков, С.В. Создание информационных систем с AllFusion Modeling Suite / С.В. Маклаков. — М.: ДИАЛОГ-МИФИ, 2005. — 432 с.
7. UML спецификация. — www.omg.com.
8. Буч, Г. Объектно-ориентированный анализ и проектирование с примерами приложений на C++ / Г. Буч. — М.: Бином, 2001. — 560 с.
9. Буч, Г. Язык UML. Руководство пользователя / Г. Буч, Дж. Рамбо, А. Якобсон. - СПб.: Питер, 2004. - 432 с.
10. Леоненков, А.В. Самоучитель UML 2 / А.В. Леоненков. — СПб.: БХВ - Петербург, 2007. — 576с.
11. Леоненков, А.В. Объектно-ориентированный анализ и проектирование с использованием UML / А.В. Леоненков. — www.intuit.ru.
12. Якобсон, А. Унифицированный процесс разработки программного обеспечения / А. Якобсон, Г. Буч, Дж. Рамбо. - СПб.: Питер, 2002. - 496 с.
13. Фаулер, М. Архитектура корпоративных программных приложений / М. Фаулер. — М.: Издательский дом «Вильямс», 2004. — 544 с.
14. Фаулер, М. UML. Основы. Третье издание. / М. Фаулер. — М.: Символ-Плюс, 2006. — 192 с.

15. Анисимов, В.В. Проектирование информационных систем. Часть 1. Структурный подход: конспект лекций / В.В. Анисимов. – Хабаровск: Изд-во ДВГУПС, 2005. – 112 с.

16. Анисимов, В.В. Проектирование информационных систем. Часть 2. Объектно-ориентированный подход: конспект лекций / В.В. Анисимов. – Хабаровск: Изд-во ДВГУПС, 2007. – 100 с.

2. Допоміжна література

1. Орлов, С.А. Технологии разработки программного обеспечения: учеб. / С.А. Орлов. – СПб.: Питер, 2002. – 464 с.

2. Маклаков, С.В. BPwin и ERwin. CASE-средства разработки информационных систем / С.В. Маклаков. – М.: ДИАЛОГ-МИФИ, 2001. – 304 с.

3. Крачтен, Ф. Введение в Rational Unified Process / Ф. Крачтен. - М.: Издательский дом «Вильямс», 2002. - 240 с.

4. Элиенс, А. Принципы объектно-ориентированной разработки программ / А. Элиенс. – М.: Издательский дом «Вильямс», 2002. – 496 с.

5. Ларман, К. Применение UML и шаблонов проектирования: Уч. Пос / К. Ларман. - М.: Издательский дом «Вильямс», 2001. - 496 с.

6. Гранд, М. Шаблоны проектирования в Java / М. Гранд. - М.: Новое знание, 2004. - 559 с.

7. Йордан, Э. Объектно-ориентированный анализ и проектирование систем / Э. Йордан, С. Аргила. - М.: Издательство «ЛОРИ», 2007. - 264 с.

8. (BPwin) Где? Зачем? Как? / В.И. Дубейковский. – М.: ДИАЛОГ-МИФИ, 2004. – 464 с.

3. Интернет-ресурси

1. Академия Microsoft: Анализ требований к автоматизированным информационным системам: <http://www.intuit.ru/studies/courses/2188/174/info>

2. АНАЛИЗ ТРЕБОВАНИЙ К ИНФОРМАЦИОННЫМ СИСТЕМАМ Конспект лекций Маглинец Ю.А. <http://ivan-shamaev.ru/wp-content/uploads/2013/06/Information-systems-analysis-and-requirements-analysis.pdf>

3. Анализ требований по Вигерсу (2004). Этапы сбора требований. <http://iiba.ru/requirements-analysis/analysis-of-requirements-wiegers-2004/> Вигерс К. Разработка требований к программному обеспечению <http://www.twirpx.com/file/1073169/>

4. С.И. Клевцов АНАЛИЗ И ФОРМИРОВАНИЕ ТРЕБОВАНИЙ К ПРОГРАММНОМУ ОБЕСПЕЧЕНИЮ ИНФОРМАЦИОННЫХ СИСТЕМ СБОРА И ОБРАБОТКИ ДАННЫХ Учебное пособие - http://rtf.sfedu.ru/!mps/umk/strdsgn_ch1.pdf

5. Проектирование микрослужбы, ориентированной на DDD <https://docs.microsoft.com/ru-ru/dotnet/architecture/microservices/microservice-ddd-cqrs-patterns/ddd-oriented-microservice>

6. Снова о разработке на основе предметной области (Domain-Driven Design, DDD) - <https://habr.com/ru/post/258693/>

7. Проектирование информационных систем - https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema8/tema8_4

8. Основные термины и определения: <https://studopedia.info/10-35578.html>

9. Диаграмма состояний (диаграмма автомата) UML: <https://planerka.info/item/diagrammy-sostoyanij-diagrammy-avtomata-uml/>

10. Архитектура распределенных систем: <https://helpiks.org/4-83847.html>

11. Архитектурные стили проектирования: https://studref.com/320288/informatika/arhitekturnye_stili_proektirovaniya

12. Многослойная архитектура: <https://studfile.net/preview/6413815/page:13/>
13. Основы архитектуры Web-приложений:
<http://studepedia.org/index.php?vol=1&post=53643>
14. Паттерны проектирования: <https://refactoring.guru/ru/design-patterns>
15. Диаграммы развертывания:
https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema15/tema15_3
16. Диаграммы компонентов
https://sites.google.com/site/anisimovkhv/learning/pris/lecture/tema15/tema15_2#p153
17. Программирование сокетов в Linux: <http://rdsn.org/article/unix/sockets.xml>
18. Технология COM:
<http://techn.sstu.ru/kafedri/подразделения/1/MetMat/murashev/com/lec/>