

Міністерство освіти і науки України  
Харківський національний університет імені В.Н. Каразіна  
Кафедра штучного інтелекту та програмного забезпечення

**“ЗАТВЕРДЖУЮ”**

Проректор  
з науково-педагогічної роботи

А.В. Пантелеймонов

08 2019 р.



**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

**Аналіз і моделювання вимог**

рівень вищої освіти	перший (бакалаврський) рівень
галузь знань	12 Інформаційні технології
спеціальність	122 Комп'ютерні науки
освітня програма	Комп'ютерні науки
вид дисципліни	обов'язкова
факультет	комп'ютерних наук

Програму рекомендовано до затвердження Вченою радою факультету комп'ютерних наук  
«27» червня 2019 року, протокол № 2

**РОЗРОБНИКИ ПРОГРАМИ:**

Старший викладач кафедри штучного інтелекту та програмного забезпечення **Горбань  
Анатолій Михайлович**

Програму схвалено на засіданні кафедри штучного інтелекту та програмного забезпечення  
Протокол від «29» травня 2019 року № 11

Завідувач кафедри штучного інтелекту та програмного забезпечення

  
\_\_\_\_\_ (Куклін В.М.)

Програму погоджено методичною комісією факультету комп'ютерних наук  
Протокол від «20» червня 2019 року № 9

Голова методичної комісії факультету комп'ютерних наук

  
\_\_\_\_\_ (Бердніков А.Г.)

## ВСТУП

Програма навчальної дисципліни «Аналіз і моделювання вимог» складена відповідно до освітньо-професійної програми підготовки першого (бакалаврського) рівня вищої освіти, спеціальності 122 «Комп'ютерні науки».

### 1. Опис навчальної дисципліни

1.1. Основною метою є надання уявлення про суть та роль системного аналізу в процесі розробки програмних продуктів; засвоєння теоретичних та практичних знань та навичок в області прикладного системного аналізу.

1.2. Формування у студентів знань в області прикладного системного аналізу в об'ємі, необхідному для його практичного використання в технологічному процесі; здобуття навиків організації та виконання відповідних процедур етапу аналізу.

Студенти мають оволодіти знаннями базових понять, принципів, методів та процедур прикладного системного аналізу.

1.3. Кількість кредитів – 4

1.4. Загальна кількість годин – 120

1.5. Характеристика навчальної дисципліни	
Нормативна / за вибором	
денна форма навчання	Заочна (дистанційна) форма навчання
Рік підготовки	
3-й	
Семестр	
6-й	
Лекції	
32 год.	_____ год.
Практичні, семінарські заняття	
32 год.	_____ год.
Лабораторні заняття	
_____ год.	_____ год.
Самостійна робота	
56 год.	_____ год.
В т.ч. індивідуальні завдання	
5 год.	_____ год..

1.6. Заплановані результати навчання

**знати:**

- базові поняття, значення та роль системного аналізу в інженерії розробки програмного забезпечення;
- принципи та методики виявлення та встановлення вимог до системи;
- принципи, методики та засоби аналізу систем;
- методики моделювання та формального опису систем.

**Вміти:**

- організувати виявлення та обробку вимог та обмежень;
- сформувати документ опису вимог відповідно заданому шаблону;
- розробити специфікації формальних моделей системи;
- організувати свою роботу у відповідності до ролі в команді розробників (аналітиків).

**2. Тематичний план навчальної дисципліни**

*Розділ 1. Предмет системного аналізу. Встановлення вимог.*

*Тема 1. Основні поняття.*

Визначення предмету системного аналізу. Основні методи аналізу систем. Особливості прикладного системного аналізу та його роль, завдання і значення в розробці ПС.

*Тема 2. Класичні методи виявлення системних і програмних вимог.*

Зміст та процедури класичних методів – інтерв'ю, анкетування, спостереження та вивчення документації.

*Тема 3. Сучасні методи виявлення вимог.*

Зміст та процедури сучасних методів – прототипування, JAD-методика та парадигми розробки на їх основі.

*Тема 4. Узгодження і перевірка обґрунтованості вимог. Документ опису вимог.*

Вимоги за рамками системи, матриця залежності вимог, ризики та пріоритети вимог, структурування вимог, управління вимогами. Структура документу опису вимог.

*Розділ 2. Класичні методи прикладного аналізу в програмній інженерії.*

*Тема 1. Структурний аналіз.*

Модель рамок системи. Діаграми потоків даних – класична модель Тома Де Марко та розширення для систем реального часу.

*Тема 2. Методи орієнтовані на структури даних.*

Метод Варньє-Орра та споріднені методи.

*Тема 3. Скінченно-автоматне зображення моделі системи.*

Поняття автомата, способи задавання, спеціальні класи автоматів. Основи теорії скінченних автоматів, їх використання в системному аналізі.

*Тема 4. Бізнес-моделі та модель прецедентів.*

Поняття та призначення прецедентної моделі системи. Бізнес-моделі в ролі моделей рамок системи.

*Розділ 3. Об'єктно-орієнтовані моделі програмних систем*

*Тема 1. Об'єктний підхід до моделювання системи. Мова моделювання UML як інструмент аналізу.*

Особливості об'єктного підходу до розробки формальної моделі системи. Уніфікована мова моделювання UML. Основні компоненти і система канонічних діаграм.

*Тема 2. Статичні моделі систем.*

Діаграма бізнес-класів – призначення, синтаксис, семантика, процедури розробки та їх роль на етапі аналізу.

*Тема 3. Динамічні моделі систем.*

Діаграми станів, діяльності, кооперації, послідовності – призначення, синтаксис, семантика, процедури розробки та їх роль на етапі аналізу.

### 3. Структура навчальної дисципліни

Назви розділів і тем	Кількість годин					
	денна форма					
	усього	у тому числі				
л		п	лаб.	Інд.	С. р.	
1	2	3	4	5	6	7
<b>Розділ 1. Предмет системного аналізу. Встановлення вимог</b>						
Тема 1. Основні поняття	6	2				4
Тема 2. Класичні методи виявлення системних і програмних вимог	10	2	4			4
Тема 3. Сучасні методи виявлення вимог	6	2				4
Тема 4. Узгодження і перевірка обґрунтованості вимог. Документ опису вимог	8	2	2			4
<b>Разом за розділом 1</b>	<b>30</b>	<b>8</b>	<b>6</b>			<b>16</b>
<b>Розділ 2. Класичні методи прикладного аналізу в програмній інженерії</b>						
Тема 1. Структурний аналіз	10	2	4			4
Тема 2. Методи орієнтовані на структури даних	6	2				4
Тема 3. Скінченно-автоматне зображення моделі системи	12	4	4			4
Тема 4. Бізнес-моделі та модель прецедентів	8	2	2			4
<b>Разом за розділом 2</b>	<b>36</b>	<b>10</b>	<b>10</b>			<b>16</b>
<b>Розділ 3. Об'єктно-орієнтовані моделі програмних систем</b>						
Тема 1. Об'єктний підхід до моделювання системи. Мова моделювання UML як інструмент аналізу	8	2				6
Тема 2. Статичні моделі систем	21	6	6			9
Тема 3. Динамічні моделі систем	25	6	10			9
<b>Разом за розділом 3</b>	<b>54</b>	<b>14</b>	<b>16</b>			<b>24</b>
<b>Усього годин</b>	<b>120</b>	<b>32</b>	<b>32</b>			<b>56</b>

#### 4. Теми практичних (лабораторних) занять

№ з/п	Назва теми	Кількість годин
<b>Розділ 1. Предмет системного аналізу. Встановлення вимог</b>		
1	Практикум з виконання процедур класичних методів виявлення вимог	4
2	Практикум з узгодження і перевірки обґрунтованості вимог.	2
<b>Розділ 2. Класичні методи прикладного аналізу в програмній інженерії</b>		
3	Розробка сценаріїв поведінки системи	4
4	Розробка моделі прецедентів	4
5	Практикум з автоматного моделювання систем	2
<b>Розділ 3. Об'єктно-орієнтовані моделі програмних систем</b>		
6	Практикум з виявлення бізнес-класів	4
7	Розробка моделі бізнес-класів	2
8	Розробка моделей кооперації та послідовності	4
9	Розробка моделі діаграми станів	2
10	Розробка моделей діяльності	4
<b>Разом:</b>		<b>32</b>

#### 5. Завдання для самостійної роботи

№ з/п	Види, зміст самостійної роботи	Кількість годин
<b>Розділ 1. Предмет системного аналізу. Встановлення вимог</b>		
1	Поглиблене ознайомлення з основами роботи з вимогами - за джерелами з списку допоміжної літератури [1] і [5].	16
<b>Розділ 2. Класичні методи прикладного аналізу в програмній інженерії</b>		
2	Самостійна розробка діаграм потоків даних, автоматних діаграм та діаграм прецедентів за завданнями.	10
<b>Розділ 3. Об'єктно-орієнтовані моделі програмних систем</b>		
3	Поглиблене ознайомлення з методами об'єктного моделювання - за джерелами з списку допоміжної літератури [4]. Самостійна розробка діаграм за завданнями.	25
4	Підготовка до контрольної роботи	5
<b>Разом</b>		<b>56</b>

#### 6. Індивідуальні завдання

Виконання практикуму з курсу передбачає виконання більшої частини проектних робіт (див. далі) в контексті самостійної роботи студентів. При цьому проробка певних проектних задач може формулюватись викладачем у вигляді індивідуальних завдань.

#### 7. Методи контролю

На протязі усього терміну викладання означеної дисципліни проводиться поточний контроль засвоєння лекційного матеріалу (контроль знань) та контроль здобуття практичних навиків (контроль вмінь). Підсумковий семестровий контроль також дозволяє контролювати як одержані знання так і набуті вміння:

- Після закінчення вивчення 1-2 розділу курсу та 3-4 розділу контроль знань проводиться у формі тестування чи контрольної роботи по матеріалах розділів.

- Після викладання курсу здійснюється підсумковий контроль у формі екзаменаційної роботи.

## 8. Схема нарахування балів

Поточний контроль та самостійна робота				Екзамен	Сума
Розділ 1 Розділ 2 Контрольна робота	Розділ 3 Розділ 4 Контрольна робота	Практикум	Разом		
10	10	40	60	40	100

### Критерії оцінювання

#### Контрольні роботи

Контрольна робота складається з 10 питань з відкритою множиною відповідей. Кожна відповідь оцінюється в 0 балів (невірна або відсутня), 0,5 бала (часткова чи неточна) чи 1 бал (вичерпна).

#### Практикум

Практичні заняття з курсу організовані як виконання ІТ проекту (в частині аналізу) групою студентів в складі до 10 осіб з зазначенням вкладу кожного члена команди в виконання етапів роботи. За результатами роботи формується фінальний звіт що включає також робочу документацію проекту – протоколи інтерв'ю, специфікації вимог, сценарії, моделі рамок системи, структур даних, бізнес-прецедентів та бізнес-класів. Фінальний звіт рецензується викладачем на відповідність стандартним (в ІТ індустрії) вимогам до проектної документації та за результатами оцінюється в обсязі від 0 до 40 балів. Оцінка кожного члена команди обраховується за формулою

$$\text{ОЦінд} = \text{ОЦзвіт} * \text{КС} * \text{КУінд} * \text{ППінд},$$

Де ОЦінд – оцінка студента за практикум (при ОЦінд >40 приймається 40), ОЦзвіт – оцінка фінального звіту (від 0 до 40), КС – кількість студентів в команді, КУінд – коефіцієнт «трудової участі» (вкладу студента в виконання проекту в цілому в діапазоні від 0 до 1) встановлюється спільно самими членами команди таким чином щоб в сумі на всю команду набиралось значення 1 (тобто 100 %), ППінд – показник присутності студента на аудиторних заняттях як відношення «був присутній» до «всього занять» (від 0 до 1).

Так, наприклад, для групи з 9 студентів при оцінці фінального звіту в 32 бали студент з КУінд = 15 % що пропустив половину занять (ППінд = 0,5) отримає оцінку ОЦінд = 32 \* 9 \* 0,15 \* 0,5 = 21,6 ≈ 22 бали, а маючи КУінд = 7 % і ППінд = 1,0 отримає ОЦінд ≈ 20 балів і, нарешті, при вкладі в спільну роботу над проектом 20 % пропустивши при цьому четверть занять матиме ОЦінд = 43,2 ≈ 40 балів.

#### Екзамен

Проводиться в письмовій формі і полягає в наданні стислих та вичерпних відповідей що не мають явних ознак плагіату на два тематичних питання вибраного білету. Орієнтовний об'єм кожної відповіді складає одну-дві рукописні сторінки. Кожна відповідь оцінюється максимум в 20 балів.

Більш детальні критерії:

Оцінка в балах	Пояснення
15 – 20	Відповідь на питання надана повністю, відмінна робота без помилок або з однією-двома незначними помилками.
10 – 14	Відповідь на питання надана повністю, робота з декількома незначними помилками, або з однією – двома значними помилками.
5 –9	Відповідь на питання надана не повністю, але прогалини не носять істотного характеру, необхідний об'єм знань матеріалу у студента в основному сформований, робота з трьома і більше значними помилками
0–4	Відповідь не надавалась або відповідь не має відношення до поставленого питання.

У випадку виявлення явних фрагментів плагіату у відповіді вони відмічаються і вважаються виключеними з тексту.

### Шкала оцінювання

Сума балів за всі види навчальної діяльності протягом семестру	Оцінка	
	для чотирирівневої шкали оцінювання	для дворівневої шкали оцінювання
90 – 100	відмінно	зараховано
70-89	добре	
50-69	задовільно	
1-49	незадовільно	не зараховано

## 9. Рекомендована література

### Основна література

1. Орлов С. А. Технология разработки программного обеспечения: Учебн. пособие. 2-е изд. - СПб: «Питер», 2003.
2. Сурмин Ю. П. Теория систем и системный анализ: Учеб. пособие. — К.: МАУП, 2003.
3. Леоненков А. Самоучитель UML: - СПб.: «БХВ-Петербург», 2004.

### Допоміжна література

1. Мацяшек Л. Анализ требований и проектирование систем. Разработка информационных систем с использованием UML: - М.: «Вильямс», 2002.
2. Брауэр В. Введение в теорию конечных автоматов:- М. Радио и связь, 1987.
3. Советов Б. Я., Яковлев С. А. Моделирование систем. – М.: Высш. школа, 3 изд., 2001.
4. Ларман К. Применение UML и шаблонов проектирования: - М.: «Вильямс», 2004.
5. Леффиигуэлл Д., Уидриг Д. Принципы работы с требованиями к программному обеспечению. – М.: «Вильямс», 2002.
6. Дж. Рамбо, А. Якобсон, Г. Буч. UML: Специальный справочник. СПб.: Питер, 2002.
7. М. Фаулер, К. Скотт. UML в кратком изложении. М., Мир, 1999.



## **10. Посилання на інформаційні ресурси в Інтернеті, відео-лекції, інше методичне забезпечення**

1. The Software Engineering Body of Knowledge: [Електроний ресурс]// IEEE, 2005-2017. URL: [http://www.swebok.org/ironman/pdf/SWEBOK\\_Guide\\_2004.pdf](http://www.swebok.org/ironman/pdf/SWEBOK_Guide_2004.pdf) .
2. F. Bachmann, L. Bass, C. Buhman, S. Comella-Dorda, F. Long, J. Robert, R. Seacord, K. Wallnau. Volume II: Technical Concepts of Component-Based Software Engineering, 2nd Edition/ Technical Report CMU/SEI-2000-TR-008: [Електроний ресурс]// Carnegie Mellon University, 2016. URL: <http://www.sei.cmu.edu/pub/documents/00.reports/pdf/00tr008.pdf>.