


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

Кафедра інженерії програмного забезпечення (№603)

ЗАТВЕРДЖУЮ

Керівник проектної групи


(підпис) І.Б. Туркін
(ініціали та прізвище)
«30» 08 2019 р.

**РОБОЧА ПРОГРАМА ОBOB'ЯЗКОВОЇ
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Технологія проектування програмного забезпечення
(назва навчальної дисципліни)

Галузь знань: 12 «Інформаційні технології»
(шифр та найменування галузі знань)

Спеціальність: 121 «Інженерія програмного забезпечення»
(код та найменування спеціальності)

Освітня програма: «Інженерія програмного забезпечення»
(найменування освітньої програми)

Форма навчання: денна

Рівень вищої освіти: перший (бакалаврський)

Харків 2019 рік

Робоча програма «Технологія проектування програмного забезпечення» для студентів за спеціальністю 121 «Інженерія програмного забезпечення» освітньої програми «Інженерія програмного забезпечення».

«20» 04 2019 р. – 13 с.

Розробник: Шевченко І.В., доц. каф. 603, канд. техн. наук., доц.
(прізвище та ініціали, посада, наукова ступінь та вчене звання)



(підпис)

Робочу програму розглянуто на засіданні кафедри інженерія програмного забезпечення

(назва кафедри)

Протокол № 1 від «26» 08 2019 р.

Завідувач кафедри

д-р техн. наук., проф.
(науковий ступінь і вчене звання)



(підпис)

І.Б. Туркін
(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 – 4.0	Галузь знань <u>12 «Інформаційні технології»</u> <small>(шифр і найменування)</small>	2. Цикл професійної підготовки (2.1. Дисципліни загально-професійної підготовки)
Кількість модулів – 2		Навчальний рік:
Кількість змістових модулів – 2	Спеціальність <u>121 «Інженерія програмного забезпечення»</u> <small>(код і найменування)</small>	2019/2020
Індивідуальне завдання: «Спеціальна частина дипломного проекту бакалавра»	Освітня програма <u>«Інженерія програмного забезпечення»</u> <small>(найменування)</small>	Семестр
Загальна кількість годин – 64/120		7-й
Кількість тижневих годин для денної форми навчання: аудиторних – 4 самостійної роботи студента – 3,5	Рівень вищої освіти: перший (бакалаврський)	Лекції*
		32 години
		Практичні, семінарські*
		Лабораторні*
		32 години
Самостійна робота		
56 годин		
Вид контролю:		
модульний контроль, іспит		

Співвідношення кількості годин аудиторних занять до самостійної і індивідуальної роботи становить: 64/56.

* Аудиторне навантаження може бути зменшене або збільшене на одну годину в залежності від розкладу занять.

2. Мета та завдання навчальної дисципліни

Мета вивчення: надати студентам знання і навички з проектування розробки програмного забезпечення (ПЗ) з врахуванням проблематики предметної області, отриманої відповідної термінології споживачів, з використанням шаблонів проектування, сучасних інформаційних технологій для розробки якісного програмне забезпечення згідно реальним потребам клієнта-замовника, витримуючи планові бюджет та термін часу.

Завдання: вивчення теоретичного матеріалу розділів, відпрацювання на лабораторних роботах та виконання розрахункової роботи щодо проектування програмного забезпечення складної системи, відповідно до потреб клієнтів-замовників з врахуванням проблематики предметної області (згідно матеріалу виробничої практики) та отриманої відповідної термінології споживачів.

Результати навчання: студент має:

знати:

- призначення, основні графічні елементи, зв'язки між елементами, приклади побудови BPM-діаграм для моделювання бізнес-процесів;
- призначення, основні графічні елементи, зв'язки між елементами, приклади побудови UML-діаграми варіантів використання;
- призначення, основні графічні елементи, зв'язки між елементами, приклади побудови UML-діаграми діяльності (активностей);
- призначення, основні графічні елементи, зв'язки між елементами, приклади побудови UML-діаграми класів;
- призначення, основні графічні елементи, зв'язки між елементами, приклади побудови UML-діаграм взаємодії: послідовності та кооперації;
- призначення, основні графічні елементи, зв'язки між елементами, приклади побудови UML-діаграм станів;
- призначення, основні графічні елементи, зв'язки між елементами, приклади побудови UML-діаграм реалізації: компонентів та розгортання;
- призначення, основні графічні елементи, зв'язки між елементами, приклади побудови моделей даних в нотації IDEF1X;
- поняття патерну і фреймворка;
- архітектурні рішення на базі шарів та про взаємодію між архітектурними шарами;
- базові патерни розробки: «Фабрика», «Спостерігач», «Інверсія контролю»;
- патерни проектування бізнес-логіки: «Модель предметної області», «Модуль таблиці», «Сценарій транзакцій»;
- патерни проектування шару сервісів: «Інтерфейс доступу до домену», «Сценарій операції»;

- патерни проектування шару інтеграції: «Шлюз таблиці даних», «Шлюз записи даних», «Активна запис», «Перетворювач даних»;
- проектування шару уявлення: архітектура тонкого клієнта, типове рішення MVC, типове рішення реалізації шару уявлення;
- поняття гнучких технологій проектування ПО, маніфест гнучкої розробки ПО, приклади гнучких технологій розробки ПЗ.

вміти:

- будувати BPM-діаграми для моделювання бізнес-процесів;
- будувати UML-діаграму варіантів використання;
- будувати UML-діаграму діяльності (активностей);
- будувати UML-діаграму класів;
- будувати UML-діаграми взаємодії: послідовності та кооперації;
- будувати UML-діаграму станів;
- будувати UML-діаграму реалізації: компонентів та розгортання.
- будувати модель даних в нотації IDEF1X.

мати уявлення:

- призначення, основні графічні елементи, зв'язки між елементами, приклади побудови діаграм у нотації IDEF0;
- призначення, основні графічні елементи, зв'язки між елементами, приклади побудови діаграм у нотації DFD;
- призначення, основні графічні елементи, зв'язки між елементами, приклади побудови діаграм у нотації IDEF3;
- про SCRUM: визначення, мета, історія, основні ролі (дійові особи), процеси, артефакти, зустрічі.
- про Canban: визначення, історія, опис методу, шість ключових практик методу.
- про XP-програмування: визначення, мета, історія, концепції: завдання і активності, основні прийоми.

Міждисциплінарні зв'язки: дисципліні передуює курс «Основи програмної інженерії», «Аналіз вимог до програмного забезпечення», «Бази даних», «Архітектура та проектування програмного забезпечення .Net», дисципліна потрібна для подальшого вивчення курсу «Тестування та верифікація програмного забезпечення (кп)».

3. Програма навчальної дисципліни

Модуль 1

Змістовий модуль 1. *Моделювання програмного забезпечення*

Тема 1. *Вступ до моделювання ПЗ*

Дипломний проект бакалавра, як кваліфікаційна робота для отримання диплома бакалавра з інженерії програмного забезпечення. Структура дипломного проекту бакалавра. Особливості комплексної розробки дипломного проекту бакалавра: ролі в проекті (системний аналітик, системний архітектор, тестувальник), виділення підсистем для кожного учасника комплексної розробки.

Методології розробки програмного забезпечення: від каскадної методології до гнучких методологій, їхні переваги й недоліки.

Тема 2. *Моделювання бізнес-процесів*

Поняття бізнес-процесу. Моделювання бізнес-процесів. Історія розвитку нотацій для бізнес-моделювання. IDEF0 – методологія функціонального моделювання. IDEF0: призначення, основні графічні елементи, зв'язки між елементами. Інструментальні засоби, що підтримують нотацію IDEF0. Приклади побудови діаграм в нотації IDEF0.

Сімейство нотацій IDEF. Нотація IDEF3 для документування технологічних процесів. Нотація DFD для моделювання потоків даних. IDEF3: призначення, основні графічні елементи, зв'язки між елементами. DFD: призначення, основні графічні елементи, зв'язки між елементами. Інструментальні засоби, що підтримують нотації IDEF3 і DFD. Приклади побудови діаграм в нотації IDEF3 і DFD.

Нотація BPM – сучасне рішення для моделювання бізнес-процесів. BPMN: призначення, основні графічні елементи, зв'язки між елементами. Інструментальні засоби, що підтримують нотацію BPM. Приклади побудови діаграм в нотації BPM.

Тема 3. *Об'єктно-орієнтоване проектування програмного забезпечення (мова UML)*

UML – уніфікована мова моделювання об'єктно-орієнтованих програмних систем. Діаграми UML 1.0 і 2.0. Класифікація UML-діаграм: поведінкові і структурні. Інструментальні засоби, що підтримують UML.

Діаграма варіантів використання: призначення, основні графічні елементи, зв'язки між елементами, приклади побудови. Діаграма діяльності: призначення, основні графічні елементи, зв'язки між елементами, приклади побудови.

Діаграма класів: призначення, основні графічні елементи, зв'язки між елементами, приклади побудови.

Діаграми взаємодії (послідовності і кооперації): призначення, основні графічні елементи, зв'язки між елементами, приклади побудови.

Діаграма станів: призначення, основні графічні елементи, зв'язки між елементами, приклади побудови.

Діаграми реалізації (компонентів і розгортання): призначення, основні графічні елементи, зв'язки між елементами, приклади побудови.

Модульний контроль

Модуль 2

Змістовний модуль 1. *Моделювання баз даних, проектування на основі патернов і фреймворків, гнучкі технології проектування*

Тема 1. *Моделювання баз даних*

Концептуальне проектування баз даних. Етапи концептуального проектування баз даних: від ER-моделі предметної області до SQL-скриптів. Інструментальні засоби, що підтримують побудову ER-моделей.

Нотація IDEF1X для проектування баз даних. Інструментальні засоби, що підтримують нотацію IDEF1X. Приклади концептуального проектування баз даних.

Тема 2. *Проектування на основі патернов і фреймворків*

Поняття патерну і фреймворка. Архітектурні рішення на базі шарів. Взаємодія між архітектурними шарами. Базові патерни розробки: «Фабрика», «Спостерігач», «Інверсія контролю». Патерни проектування бізнес-логіки: «Модель предметної області», «Модуль таблиці», «Сценарій транзакцій». Патерни проектування шару сервісів: «Інтерфейс доступу до домену», «Сценарій операції». Патерни проектування шару інтеграції: «Шлюз таблиці даних», «Шлюз записи даних», «Активна запис», «Перетворювач даних». Проектування шару уявлення: архітектура тонкого клієнта, типове рішення MVC, типове рішення реалізації шару уявлення.

Тема 3. *Гнучкі технології проектування програмного забезпечення*

Поняття гнучких технологій проектування ПО. Маніфест гнучкої розробки ПО. Приклад гнучких технологій розробки ПЗ: історія їх розвитку.

SCRUM: визначення, мета, історія, основні ролі (дійові особи), процеси, артефакти, зустрічі.

Scaban: визначення, історія, опис методу, шість ключових практик методу.

XP-програмування: визначення, мета, історія, концепції: завдання і активності, основні прийоми.

Модульний контроль

4. Структура навчальної дисципліни

Назва змістовного модуля і тем	Кількість годин				
	Усього	У тому числі			
		л	п	лаб.	с. р.
1	2	3	4	5	6
Модуль 1					
<i>Змістовний модуль 1. Моделювання програмного забезпечення</i>					
Тема 1. Вступ до моделювання ПЗ	6	4			2
Тема 2. Моделювання бізнес-процесів	18	4		10	4
Тема 3. Об'єктно-орієнтоване проектування програмного забезпечення (мова UML)	24	8		12	4
Модульний контроль	5				5
Разом за змістовним модулем 1	53	16		22	15
Усього годин	53	16		22	15
Модуль 2					
<i>Змістовний модуль 1. Моделювання баз даних, проектування на основі патернов і фреймворків, гнучкі технології проектування</i>					
Тема 1. Моделювання баз даних	12	4		4	4
Тема 2. Проектування на основі патернов і фреймворків	16	4		6	6
Тема 3. Гнучкі технології проектування програмного забезпечення	14	8			6
Модульний контроль	5				5
Разом за змістовним модулем 1	47	16		10	21
Усього годин	47	16		10	21
Індивідуальне завдання	10				10
Контрольний захід	10				10
Усього годин	120	32		32	56

5. Теми семінарських занять

№ п/п	Назва теми	Кількість годин
1		
2		
	Разом	

6. Теми практичних занять

№ п/п	Назва теми	Кількість годин
1		
2		
	Разом	

7. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Обґрунтування вибору теми дипломного проекту бакалавра. Мета і загальні завдання на дипломний проект бакалавра.	2
2	Моделювання бізнес-процесів.	4
3	Вимоги замовника. Архітектура програмного забезпечення. План системного тестування.	6
4	Аналіз аналогів програмного забезпечення. Побудова та опис діаграми класів, діаграми компонентів і діаграми розгортання. План інтеграційного тестування.	6
5	Вимоги до програмного забезпечення. Специфікація варіантів використання. Обґрунтування вибору мови програмування, інструментального середовища розробки і СУБД. Специфікації системних тестів.	6
6	Розробка екранних форм. Проектування моделі даних. Специфікації системних тестів.	6
	Разом	32

8. Самостійна робота

№ з/п	Назва теми	Кількість годин
1	Вступ до моделювання ПЗ	8
2	Моделювання бізнес-процесів	12
3	Об'єктно-орієнтоване проектування програмного забезпечення (мова UML)	16
4	Моделювання баз даних	12
5	Проектування на основі патернів і фреймворків	12
6	Гнучкі технології проектування програмного забезпечення	12
7	Виконання індивідуального завдання	10

8	Підготовка до контрольних заходів	10
	Разом	92

9. Індивідуальні завдання

Виконання розрахункової роботи за індивідуальним варіантом за темою « Спеціальна частина дипломного проекту бакалавра».

10. Методи навчання

За джерелами придбання знань – словесні: лекція (вступна, традиційна, проблемна, з помилками), бесіда (евристична), диспут, дискусія, робота з друкованими та інтернет-джерелами; наочні: ілюстрація, спостереження; практичні: вправа, лабораторна робота.

За характером пізнавальної діяльності тих, хто навчається – інформаційно-репродуктивний, репродуктивний, проблемне викладання, частково-пошуковий.

За логікою пізнання – індуктивний, дедуктивний, аналогій, вивідних знань.

Методи перевірки й оцінки знань, умінь, навичок: спостереження, усне опитування, контрольні роботи, програмований контроль, тестування (традиційне та машинне).

11. Методи контролю

Опитування на лекціях. Виконання і захист лабораторних робіт. Модульні контрольні роботи.

Форма підсумкового контролю успішності навчання: іспит (письмово).

12. Критерії оцінювання та розподіл балів, які отримують студенти

12.1. Розподіл балів, які отримують студенти (кількісні критерії оцінювання)

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Змістовний модуль 1			
Робота на лекціях			
Виконання і захист лабораторних (практичних) робіт	5...10	3	15..30
Модульний контроль	12...15	1	11..15

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Змістовний модуль 2			
Робота на лекціях			
Виконання і захист лабораторних (практичних) робіт	5...10	3	15..30
Модульний контроль	12...15	1	11...15
Виконання і захист РГР (РР, РК)	8..10	1	8..10
Усього за семестр			60...100

Семестровий контроль (іспит) проводиться у разі відмови студента від балів поточного тестування й за наявності допуску до іспиту. Під час складання семестрового іспиту студент має можливість отримати максимум 100 балів.

Білет для іспиту/заліку складається з двох теоретичних питань (кожне питання 25 балів) та двох практичних питань (кожне питання 25 балів).

12.2. Якісні критерії оцінювання

Необхідний обсяг знань для одержання позитивної оцінки: призначення, основні графічні елементи, зв'язки між елементами, приклади побудови BPM-діаграм для моделювання бізнес-процесів; призначення, основні графічні елементи, зв'язки між елементами, приклади побудови UML-діаграми варіантів використання; призначення, основні графічні елементи, зв'язки між елементами, приклади побудови UML-діаграми діяльності (активностей); призначення, основні графічні елементи, зв'язки між елементами, приклади побудови UML-діаграми класів; призначення, основні графічні елементи, зв'язки між елементами, приклади побудови UML-діаграм взаємодії: послідовності та кооперації; призначення, основні графічні елементи, зв'язки між елементами, приклади побудови UML-діаграм станів; призначення, основні графічні елементи, зв'язки між елементами, приклади побудови UML-діаграм реалізації: компонентів та розгортання; призначення, основні графічні елементи, зв'язки між елементами, приклади побудови моделей даних в нотації IDEF1X; поняття патерну і фреймворка; архітектурні рішення на базі шарів та про взаємодію між архітектурними шарами; базові патерни розробки: «Фабрика», «Спостерігач», «Інверсія контролю»; патерни проектування бізнес-логіки: «Модель предметної області», «Модуль таблиці», «Сценарій транзакцій»; патерни проектування шару сервісів: «Інтерфейс доступу до домену», «Сценарій операції»; патерни проектування шару інтеграції: «Шлюз таблиці даних», «Шлюз записи даних», «Активна запис», «Перетворювач даних»; проектування шару уявлення: архітектура тонкого клієнта, типове рішення MVC, типове рішення реалізації

шару уявлення; поняття гнучких технологій проектування ПО, маніфест гнучкої розробки ПО, приклади гнучких технологій розробки ПЗ.

Необхідний обсяг вмінь для одержання позитивної оцінки: будування BPM-діаграми для моделювання бізнес-процесів; будування UML-діаграму варіантів використання; будування UML-діаграму діяльності (активностей); будування UML-діаграму класів; будування UML-діаграми взаємодії: послідовності та кооперації; будування UML-діаграму станів; будування UML-діаграму реалізації: компонентів та розгортання; будування модель даних в нотації IDEF1X.

12.3 Критерії оцінювання роботи студента протягом семестру

Задовільно (60-74). Здати всі лабораторні роботи (зі значною корекцією тексту лабораторної роботи викладачем), захистити розрахункову роботу (зі значною корекцією тексту розрахункової роботи викладачем), мати необхідних мінімум знань за всіма темами та мінімум вмінь щодо застосування отриманих знань.

Добре (75-89). Здати всі лабораторні роботи (з мінімальною корекцією тексту лабораторної роботи викладачем), захистити розрахункову роботу (з мінімальною корекцією тексту розрахункової роботи викладачем), знати всі теми та уміти застосовувати їх.

Відмінно (90-100). Здати всі лабораторні роботи (без корекції тексту лабораторної роботи викладачем), захистити розрахункову роботу (без корекції тексту розрахункової роботи викладачем), досконально знати всі теми та уміти застосовувати їх.

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

13. Методичне забезпечення

1. Шевченко І.В. Проектування програмного забезпечення. Основи побудови UML-діаграм [Електронний ресурс] : навч. посіб. / І. В. Шевченко, Ю. А. Кузнецова. – Харків : Нац. аерокосм. ун-т ім. М. С. Жуков-ського "Харків. авіац. ін-т", 2019. – 82 с.
2. Шевченко И.В. Моделирование информационных систем: учеб. пособие / И.В. Шевченко. – Х.: Нац. аэрокосм. ун-т «Харьк. авиац. ин-т», 2008. – Ч. 1: ВРwin, ERwin. – 75 с.

3. Шевченко И.В. Моделирование информационных систем: учеб. пособие / И.В. Шевченко, А.С. Топал. – Х.: Нац. аэрокосм. ун-т «Харьк. авиац. ин-т», 2009. – Ч. 2: UML. Rational Rose. – 81 с.

14. Рекомендована література

Базова

1. Боггс, У. UML и Rational Rose 2002 / У. Боггс, М. Боггс. – М.: Лори, 2004. – 528 с.
2. Буч, Г. UML. Руководство пользователя / Г. Буч, Дж. Рамбо, А. Джекобсон. – М.: ДМК, 2000. – 432 с.
3. Дудзяний, І. М. Об'єктно-орієнтоване моделювання програмних систем: навч. посіб. – Львів: Видавничий центр ЛНУ імені Івана Франка, 2007. – 108 с.
4. Кватрани, Т. Rational Rose 2000 и UML. Визуальное моделирование / Т. Кватрани. – М.: ДМК Пресс, 2001. – 176 с.
5. Ларман, К. Применение UML 2.0 и шаблонов проектирования / К. Ларман. – М.: ООО «Вильямс», 2007. – 736 с.
6. Леоненков, А. В. Самоучитель UML / А. В. Леоненков. – СПб.: ВHV, 2004. – 432 с.
7. Казимир В. В. Технологии проектирования программных систем / В. В. Казимир, Ю. Н. Каравайченко, В. В. Литвинов, С. Н. Шкарлет. – Чернигов: Черниговский гос. ун-т, 2013. – 363 с. – Режим доступа: <https://issuu.com/sveka/docs/monogr7>

Допоміжна

1. Трофимов, С. А. CASE-технологии: практическая работа в Rational Rose / С. А. Трофимов. – М.: Бином-Пресс, 2002. – 288 с.
2. Фаулер, М. UML. Основы / М. Фаулер, К. Скотт. – СПб.: Символ-Плюс, 2002. – 192 с.
3. Федотова, Д. Э. CASE-технологии: практикум / Д. Э. Федотова, Ю. Д. Семенов, К. Н. Чижик. – М.: Горячая линия-Телеком, 2005. – 160 с.
4. Шмуллер, Дж. Освой самостоятельно UML за 24 часа / Дж. Шмуллер. – М.: Изд. дом «Вильямс», 2002. – 352 с.
5. Владимир Грекул. Проектирование информационных систем. – Режим доступа: <https://www.intuit.ru/studies/courses/2195/55/info>

15. Інформаційні ресурси

<https://www.uml.org/>

<https://erwin.com/>

<https://www.scrum.ua/materials>

<https://habr.com/ru/post/64997/>

<https://worksection.com/blog/extreme-programming.html>

<https://habr.com/ru/post/193136/>

<https://metanit.com/sql/tutorial/>

<https://eax.me/database-design/>