

Додаток до робочої програми навчальної дисципліни
«Об'єктно-орієнтоване програмування»

Дію робочої програми продовжено: на 2021/2022 н. р.

Заступник декана факультету комп'ютерних наук з навчальної роботи



Євгенія КОЛОВАНОВА

«25» червня 2021 р.

Голова науково-методичної комісії факультету комп'ютерних наук



Анатолій БЕРДНІКОВ

«25» червня 2021 р.

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

“ЗАТВЕРДЖУЮ”
Проректор
з науково-педагогічної роботи
Антон ПАНТЕЛЕЙМОНОВ
«_____» _____ 2020 р.



РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Об'єктно-орієнтоване програмування

рівень вищої освіти	<u>перший (бакалаврський) рівень</u>
галузь знань	<u>12 Інформаційні технології</u>
спеціальність	<u>123 Комп'ютерна інженерія</u>
освітня програма	<u>Комп'ютерна інженерія</u>
вид дисципліни	<u>обов'язкова</u>
факультет	<u>комп'ютерних наук</u>

Програму рекомендовано до затвердження Вченою радою факультету комп'ютерних наук
«31» серпня 2020 року, протокол № 12

РОЗРОБНИК ПРОГРАМИ:

доцент кафедри штучного інтелекту та програмного забезпечення **Поклонський Євген
Васильович**

Програму схвалено на засіданні кафедри штучного інтелекту та програмного забезпечення
Протокол від «31» серпня 2020 року № 1

Завідувач кафедри штучного інтелекту та програмного забезпечення


Володимир КУКЛІН

Програму погоджено з гарантом освітньої програми «Комп'ютерна інженерія»

Гарант освітньої програми «Комп'ютерна інженерія»


Сергій ШМАТКОВ

Програму погоджено методичною комісією факультету комп'ютерних наук

Протокол від «31» серпня 2020 року № 1

Голова методичної комісії факультету комп'ютерних наук


Анатолій БЕРДНІКОВ

ВСТУП

Програма навчальної дисципліни «Об'єктно-орієнтоване програмування» складена відповідно до освітньо-професійної програми підготовки першого (бакалаврського) рівня вищої освіти за спеціальністю: 123 Комп'ютерна інженерія.

1. Опис навчальної дисципліни

1.1. Метою вивчення курсу «Об'єктно-орієнтоване програмування» є навчити студентів самостійно розробляти невеликі програмні продукти за допомогою об'єктно-орієнтованого підходу.

1.2. Основними завданнями вивчення дисципліни є:

- навчитися використовувати об'єктно-орієнтований підхід при аналізі явищ та систем;
- досягти рівня вільного володіння переважною більшістю синтаксичних конструкцій мови Java;
- набути практичних навичок створення та відлагодження програмного кода з використанням мови Java та супутніх програмних засобів (компілятор, IDE тощо);
- познайомитися на практиці з найчастіше використовуваними пакетами платформи J2SE та отримати уявлення про наявність інших пакетів;
- познайомитися на практиці із методами аналізу, проектування ПЗ та методологіями розробки програмних систем.

В ході вивчення дисципліни у студента повинні формуватися наступні компетентності.

Інтегральна компетентність.

Здатність розв'язувати складні задачі та вирішувати практичні завдання під час професійної діяльності в комп'ютерній галузі, що передбачає застосування теорій та методів інформаційних технологій і характеризуються комплексністю та невизначеністю умов і вимог.

Загальні компетентності (ЗК).

- Здатність до абстрактного мислення, аналізу і синтезу (ЗК 1).

Фахові компетентності спеціальності (ФК)

- Здатність застосовувати законодавчу та нормативно-правову базу, а також державні та міжнародні вимоги, практики і стандарти з метою здійснення професійної діяльності в галузі комп'ютерної інженерії. (ФК 1).
- Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж (ФК 3).

1.3. Кількість кредитів – 5

1.4. Загальна кількість годин – 150

1.5. Характеристика навчальної дисципліни	
Нормативна / за вибором	
денна форма навчання	денна форма навчання
Рік підготовки	
2-й,	-й
Семестр	
1-й	-й
Лекції	
32 год.	_____ год.
Практичні, семінарські заняття	
_____ год.	_____ год.
Лабораторні заняття	
32 год.	_____ год.
Самостійна робота	
86 год.	_____ год.
В т.ч. індивідуальні завдання	
45 год.	

1.6. Заплановані результати навчання

знати:

- сучасні технологій та інструментальні засоби розробки програмних систем, уміння їх застосовувати на всіх етапах життєвого циклу;
- розуміння основних індустріальних мов програмування, інформаційних систем і WEB.

вміти:

- об'єктно-орієнтовано мислити, застосовувати знання об'єктно-орієнтованих мов програмування та об'єктно-орієнтований підхід при проектуванні складних програмних систем.

В результаті вивчення дисципліни у студента повинні формуватися наступні програмні результати навчання (ПРН).

- ПРН 2.Мати навички проведення експериментів, збирання даних та моделювання в комп'ютерних системах.
- ПРН 6.Вміти застосовувати знання для ідентифікації, формулювання і розв'язування технічних задач спеціальності, використовуючи методи, що є найбільш придатними для досягнення поставлених цілей.
- ПРН 7.Вміти розв'язувати задачі аналізу та синтезу засобів, характерних для спеціальності.
- ПРН 8.Вміти системно мислити та застосовувати творчі здібності до формування нових ідей.
- ПРН 9.Вміти застосовувати знання технічних характеристик, конструктивних особливостей, призначення і правил експлуатації програмно-технічних засобів комп'ютерних систем та мереж для вирішення технічних задач спеціальності.
- ПРН 13.Вміти ідентифікувати, класифікувати та описувати роботу комп'ютерних систем та їх компонентів.
- ПРН 17.Спілкуватись усно та письмово з професійних питань українською мовою та однією з іноземних мов (англійською, німецькою, італійською, французькою, іспанською).
- ПРН 18.Використовувати інформаційні технології та для ефективного спілкування на професійному та соціальному рівнях.

- ПРН 19.Здатність адаптуватись до нових ситуацій, обґрунтовувати, приймати та реалізовувати у межах компетенції рішення.
- ПРН 20.Усвідомлювати необхідність навчання впродовж усього життя з метою поглиблення набутих та здобуття нових фахових знань, удосконалення креативного мислення.
- ПРН 21.Якісно виконувати роботу та досягати поставленої мети з дотриманням вимог професійної етики.

2. Тематичний план навчальної дисципліни

Розділ 5. Потоки.

Тема 16. Програмування множинних потоків. Огляд курсу

Багато поточне програмування. Модель потоків в Java. Головний потік, створення потоку. Створення множинних потоків, пріоритети потоків. Міжпоточні зв'язки.

Тема 17. Робота з потоками.

Синхронізація. Внутрішні блокування. Живучість (Liveness): взаємне блокування (Deadlock), голодування (starvation), активна локіровка (livelock). Захищені блокування (guarded blocks). Незмінні об'єкти (immutable objects).

Тема 18. Введення / виведення в Java.

Потоки введення / виведення. Клас File. Байтові потоки. Символьні потоки. Робота з файлами з довільним доступом. Серіалізація.

Розділ 6. AWT, Swing. Введення в проектування

Тема 19. Основи проектування.

Мова UML та case засоби. Діаграми UML. Варіанти використання і дійові особи.

Тема 20. Основи об'єктно-орієнтованого проектування.

Взаємодія об'єктів. Класи і пакети. Атрибути, операції,

Тема 21. Генерація програмного коду.

Підготовка до генерації програмного коду. Генерація програмного коду Java. Зворотне проектування програмного коду Java.

Тема 22. Основи AWT: робота з вікнами.

Основи віконної графіки. Робота з фреймової вікнами. Малювання простих об'єктів.

Тема 23. 8 Обробка подій: модель делегування подій.

Події та їх джерела. Інтерфейси слухачів. Класи-адаптери.

Тема 24. Основи AWT: робота з текстом.

Робота з кольором. Робота зі шрифтом. Клас FontMetrics. Основи AWT: елементи управління (Кнопки. Списки. Панелі меню), використання менеджерів компоновання (BorderLayout, FlowLayout, GridLayout, CardLayout).

Тема 25. Аплети.

Загальні відомості про аплетах. . Методи алетів, висновок графіки. Питання безпеки.

Тема 26. Робота з бібліотекою Swing.

Клас JFrame. Створення стандартних елементів управління Swing (кнопки і їх групи, радиокнопки, прапорці, списки, що випадають і ін.). Бібліотека Swing: каталог компонентів: текстові поля, бордюри, панелі скролінгов, списки що випадають та ін.

Тема 27. Робота з бібліотекою Swing.

Каталог компонентів: меню, спливаючі меню, вікна діалогу, файлові діалоги. Елементи GUI і пов'язані з ними події. Модель проектування MVC. Порівняння побудови програми на основі MVC та без нього на прикладі.

Тема 28. Огляд Java FX.

Огляд Java FX (технологія Java Beans, BDK, інтроспекція).

3. Структура навчальної дисципліни

Назви розділів і тем	Кількість годин								
	денна форма								
	усього	у тому числі							
		лекц.		прак.		лаб.		інд.	с.р.
ауд.	дист.	ауд.	дист.	ауд.	дист.				
1	2	3	4	5	6	7	8	9	10
Розділ 5. Потoki.									
Тема 16. Програмування множинних потоків. Огляд курсу	10		2			2		3	3
Тема 17. Робота з потоками.	16		4			4		5	3
Тема 18. Введення / виведення в Java.	16		4			4		5	3
Разом за розділом 5	42		10			10		13	9
Розділ 6. AWT, Swing. Введення в проектування									
Тема 19. Основи проектування.	9		2			2		2	3
Тема 20. Основи об'єктно-орієнтованого проектування.	16		4			4		5	3
Тема 21. Генерація програмного коду.	9		2			2		2	3
Тема 22. Основи AWT: робота з вікнами.	9		2			2		2	3
Тема 23. Обробка подій: модель делегування подій.	11		2			2		4	3
Тема 24. Основи AWT: робота з текстом.	9		2			2		2	3
Тема 25. Аплети.	11		2			2		4	3
Тема 26. Робота з бібліотекою Swing.	13		2			2		5	4
Тема 27. Робота з бібліотекою Swing.	11		2			2		4	3
Тема 28. Обзор Java FX.	10		2			2		2	4
Разом за розділом 6	108		22			22		32	32
<i>Усього годин</i>	<i>150</i>		<i>32</i>			<i>32</i>		<i>45</i>	<i>41</i>

4. Теми практичних (лабораторних) занять

№ з/п	Назва теми	Кількість годин
Розділ 5. Потоки.		
1	Багатопотокове програмування. Робота з декількома потоками.	4
2	Потоки вводу.вивода	4
3	Серілізація.	4
Розділ 6. AWT, Swing. Введення в проектування		
4	Розробка програми з використанням AWT	4
5	Розробка програми з використанням Swing	8
6	Розробка аплету	8
	Разом	32

5. Завдання для самостійної роботи

№ з/п	Види, зміст самостійної роботи	Кількість годин
Розділ 5. Потоки.		
1	Багатопотокове програмування. Робота з декількома потоками. Самостійне опрацювання лекційного матеріалу. Шилдт[3] (глава 11, 27). Контроль: опитування під час практичного заняття, опитування під час прийому лабораторної роботи № 1 . Підготовка до контрольної роботи	3
2	Потоки вводу.вивода. Робота з декількома потоками. Самостійне опрацювання лекційного матеріалу. Шилдт[3] (глава 19). Контроль: опитування під час практичного заняття	3
3	Серілізація. Самостійне опрацювання лекційного матеріалу. Контроль: опитування під час практичного заняття, опитування під час прийому лабораторної роботи № 2	3
Розділ 6. AWT, Swing. Введення в проектування		
4	Розробка програми з використанням AWT. Самостійне опрацювання лекційного матеріалу. Шилдт[3] (глави 23,24,25). Контроль: опитування під час практичного заняття, , опитування під час прийому лабораторних робіт № 3,4.	10
5	Розробка програми з використанням Swing. Самостійне опрацювання лекційного матеріалу. Шилдт[3] (глави 26,30). Контроль: опитування під час практичного заняття, , опитування під час прийому лабораторних робіт № 5,6.	10
6	Розробка аплету. Самостійне опрацювання лекційного матеріалу. Шилдт[3] (глави 22). Контроль: опитування під час практичного заняття, опитування під час прийому лабораторної роботи № 7.	12
	Курсовий проект	45
	Разом	86

6. Індивідуальні завдання

Індивідуальна робота складається з виконання лабораторних робіт з використанням мови Java згідно методичним вказівкам та курсової роботи.

Курсова робота. Як наукове та навчально-дослідне завдання кожен студент має виконати курсову роботу. В ході роботи над проектом студент повинен показати розуміння основ ООП та основ ООП проектування, графічного інтерфейсу, концепції MVC, роботи з файлами. Студент повинен розробити програму, пояснювальну записку до неї, створити презентацію та захистити свою роботу перед комісією та студентами. Кожному студенту дається можливість самостійно вибрати тему для виконання. Студент повинен обговорити тему роботи з викладачем та затвердити її.

В якості теми роботи може бути, наприклад

- 1) гра (сапер, змійка, морський бій і т.п.)
- 2) термінал для роботи з «квазіБД» (реєстратура, бібліотека і т.п.) (зберігання даних в файлах)
- 3) тренажер (запам'ятовування іноземних слів, тестування з якогось предмету і т.п.)

7. Методи контролю

У третьому семестрі передбачений захист лабораторних робіт перед викладачем, індивідуальної роботи, а також спільний проект, при виконанні якого прогрес контролює викладач. Результати роботи над проектом захищаються. Програмні продукти, розроблені відповідно темам курсових робіт, на CD диску та пояснювальні записки у вигляді паперових звітів зберігаються на кафедрі.

Екзамен в кінці семестра складається з теста та практичного завдання.

8. Схема нарахування балів

Поточний контроль, самостійна робота, індивідуальні завдання										Екзамен	Сума
ЛР 1	ЛР 2	ЛР 3	ЛР 4	ЛР 5	ЛР 6	ЛР 7	Індивідуальне завдання (курсовий проект)	Контрольна робота, передбачена навчальним планом	Разом		
5	5	5	5	5	3	2	20	10	60	40	100

КРИТЕРІЇ ОЦІНКИ УСПІШНОСТІ РЕЗУЛЬТАТІВ НАВЧАННЯ З КУРСУ «Об'єктно-орієнтоване програмування»

<i>Форми навчальної діяльності</i>	<i>Балів за семестр</i>
1. Виконання практичних завдань	18
2. Виконання лабораторних робіт	12
3. Виконання контрольної роботи	10
4. Курсова робота	20
5. Екзамен (письмова робота)	40
Разом балів	100

Виконання лабораторних робіт

Протягом семестру студент має виконати 3 лабораторні роботи за темами:

Теми лабораторних робіт	Термін виконання	Кількість балів
Лабораторна робота №1 Багатопотоковість.	2 тижні	4
Лабораторна робота №2 Введення/виведення даних. Робота з файлами.	2 тижні	4
Лабораторна робота №3 Графічний за стосунок з анімацією.	2 тижні	4
Всього		12

Критерії оцінювання лабораторних робіт

Студент вивчає вимоги та технічне завдання, виконує роботу самостійно. Якщо програмний код робіт різних студентів має суттєву ступінь схожості, то ці роботи дискваліфікуються та не оцінюються.

Захист лабораторної роботи складається з двох етапів.

Перший етап – представлення результатів: готового програмного додатку та програмних кодів у відповідності до завдання лабораторної роботи.

Критерії оцінювання результатів роботи:

- робота була виконана у відповідності з технічним завданням за вказаний час – студент отримує 100% від максимальної кількості балів;
- робота була виконана у відповідності з технічним завданням с запізненням – студент отримує 50 % від максимальної кількості балів;
- технічне завдання виконано не повністю, а на x % – студент отримує $x\%$ від балів, що мали б бути зараховані у відповідності до строків виконання.

Другий етап здачі лабораторної роботи – відповідь на контрольні питання. Кількість питань визначає викладач за результатами представлених результатів, але не менше ніж 2. Кожне контрольне питання оцінюється таким чином:

- повна розгорнута відповідь з прикладами та додатковим завданням, що було опрацьовано на самостійній роботі – кількість балів, що отримані на першому етапі збільшується до 25 % балів;
- повна, але не розгорнута відповідь без додаткового завдання – кількість балів не змінюється;
- неповна відповідь, або відповідь, що містить незначні та некритичні помилки чи суперечності – кількість балів, що було отримано на попередньому етапі зменшується на 25 %;
- відповідь, що містить критичні помилки, або відсутність відповіді – кількість балів, що було отримано на попередньому етапі зменшується на 50 %.

Виконання практичних завдань

Протягом семестру студент має виконати 13 практичних завдань за темами:

Теми практичних занять	Вид завдання	Кількість балів
Практичне заняття №1 Створення множинних потоків, пріоритети потоків. Міжпоточні зв'язки.	практичне	2
Практичне заняття №2 Синхронізація. Взаємні блокування.	практичне	2
Практичне заняття №3 Потоки введення/виведення.	практичне	2
Практичне заняття №4 Основи віконної графіки. Робота з фреймової вікнами.	практичне	2
Практичне заняття №5 Події та їх джерела. Інтерфейси слухачів. Класи-адаптери.	практичне	2
Практичне заняття №6 Основи AWT: елементи управління, використання менеджерів компоунування.	практичне	2
Практичне заняття №7 Бібліотека Swing.	практичне	2
Практичне заняття №8 Каталог компонентів: меню, спливаючі меню, вікна діалогу, файлові діалоги. Елементи GUI і пов'язані з ними події.	практичне	2
Практичне заняття №9 Модель проектування MVC. Порівняння побудови програми на основі MVC та без нього на прикладі.	практичне	2
Всього		18

Критерії оцінювання практичних завдань

Перед виконанням студент вивчає вимоги завдання, відбувається обговорення, щоб завдання було зрозуміле коректно. Кожен студент виконує завдання самостійно. Дозволяється сумісне обговорення, але програмний код кожен студент пише власноруч.

Захист практичного завдання складається з представлення результатів: готового програмного додатку та програмних кодів у відповідності до завдання.

Виконання контрольної роботи

Протягом семестру студент має виконати письмову контрольну роботу. Контрольна робота виконується в аудиторії, розраховані на 80 хвилин та оцінюються у 10 балів.

Контрольні роботи складаються з практичних та теоретичних завдань. Кожне завдання має фіксовану максимальну кількість балів:

Список завдань	Тип завдання	Кількість балів
Завдання 1.	теоретичне	2
Завдання 2.	теоретичне	2
Завдання 3.	теоретичне	1
Завдання 4.	практичне	5

Критерії оцінювання контрольних робіт

Теоретичні завдання оцінюються таким чином:

- 100% від запланованої кількості балів – у випадку вірної відповіді;
- 80% від запланованої кількості балів – у випадку незначної неточності у відповіді;
- 50% від запланованої кількості балів – у випадку декількох незначних неточностей у відповіді;
- 0 балів – у випадку значної неточності, помилки, або відсутності відповіді.

Практичні завдання оцінюються таким чином:

- 100% від запланованої кількості балів у випадку готового програмного додатку та програмних кодів у відповідності до завдання;
- 0 балів – у випадку відсутності діючого програмного кода.

Критерії оцінювання курсового проекту

Курсовий проект оцінюється сумарно в 20 балів. Оцінка складається з таких елементів:

Критерій	Бали
Документація	5
Проектування	5
Тестування функціональних вимог	5
Якість презентації	3
Доклад та відповіді на запитання	2
Всього	20

Екзамен

Екзамен проводиться в письмовій формі, розрахован на 3 години, оцінюються у 40 балів.

Екзаменаційна робота складається з практичних та теоретичних завдань. Кожне завдання має фіксовану максимальну кількість балів:

Список завдань	Тип завдання	Кількість балів
Завдання 1.	теоретичне	10
Завдання 2.	теоретичне	10
Завдання 3.	практичне	20
Всього		40

Критерії оцінювання екзамену

Теоретичні завдання оцінюються таким чином:

- 100% від запланованої кількості балів – у випадку вірної відповіді;
- 80% від запланованої кількості балів – у випадку незначної неточності у відповіді;
- 50% від запланованої кількості балів – у випадку декількох незначних неточностей у відповіді;
- 0 балів – у випадку значної неточності, помилки, або відсутності відповіді.

Практичні завдання оцінюються таким чином:

- 100% від запланованої кількості балів у – випадку готового програмного додатку та програмних кодів у відповідності до завдання;
- 0 балів – у випадку відсутності діючого програмного кода.

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. Horstmann C. S. Core Java Volume I – Fundamentals: 11th Edition, Pearson 2018, 928 p.
2. Horstmann C. S. Core Java, Volume II--Advanced Features: 11th Edition, Pearson 2019, 1040p.
3. Schildt H. Java: A Beginner's Guide: 8th Edition, McGraw-Hill Education, 2018, 684 p.
4. Васильєв О. Програмування мовою Java: Навчальна книга Богдан 2020, 696с.
5. Копитко М.Ф., Іванків К.С. Основи програмування мовою Java: Тексти лекцій. – Львів: Видавничий центр ЛНУ ім. Івана Франка, 2002. – 83 с.
6. Брнакевич І.Є., Вагін П.П. Програмування мовою Java: використання фундаментальних класів: Тексти лекцій. – Львів: Видавничий центр ЛНУ імені Івана Франка, 2002. – 75 с.

Допоміжна література

1. Eckel B. Thinking in Java 4th Edition: Pearson, 2006, 1150 p.
2. Bloch J. Effective Java: 3rd Edition, Addison Wesley, 2017, 412 p.
3. Deitel P., Deitel H. Java How to Program, Early Objects: 11th Edition, Pearson, 2017, 1296 p.
4. Deitel P., Deitel H. Java How To Program, Late Objects: 11th Edition, Pearson, 2017, 1248 p

Посилання на інформаційні ресурси в Інтернеті, відео-лекції, інше методичне забезпечення

1. Іванов Л. В. Основи програмування Java // URL: http://iwanoff.inf.ua/java_ua
2. Освоюємо Java – Вікіпідручник // http://uk.wikibooks.org/wiki/Освоюємо_Java
3. Java Підручник. Уроки для початківців. W3Schools українською// <https://w3schoolsua.github.io/java/index.html>
4. Java Tutorial // <https://www.w3schools.com/java/>
5. Bruce Eckel. Thinking in Java, 4th Edition // http://sd.blackball.lv/library/Thinking_in_Java_4th_edition.pdf
6. Java tutorials: [Електроний ресурс]// Oracle and/or its affiliates, 1994-2017. URL: <http://docs.oracle.com/javase/tutorial/reallybigindex.html>
7. Java™ Platform, Standard Edition 8 API Specification: [Електроний ресурс]// Oracle and/or its affiliates, 1993- 2017. URL:<http://docs.oracle.com/javase/8/docs/api/>