

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Харківський національний автомобільно-дорожній університет

Група МК-21,МП-21

ЗАТВЕРДЖУЮ
Перший проректор з НІП
професор С.Я. Ходирєв
“ 8 09 2020 року



ЕХ

РОБОЧА ПРОГРАМА

навчальної дисципліни Об'єктно-орієнтоване програмування
(назва навчальної дисципліни згідно освітньої програми)

підготовки бакалавра
(назва освітньо-кваліфікаційного рівня)

в галузі знань 12 Інформаційні технології
(шифр і назва галузі знань)

спеціальності 121 Інженерія програмного забезпечення
122 Комп'ютерні науки
(шифр і назва спеціальності)

за освітньою програмою¹ Програмне забезпечення систем
Інформаційні управляючі системи і технології
(назва освітньо-професійної (освітньо-наукової) програми)

мова навчання державна
(мова, на якій проводиться навчання за робочою програмою)

2020 рік

¹ якщо програма навчальної дисципліни розроблена для декількох освітніх програм за даною спеціальністю, то вказуються усі освітні програми

1. Метою вивчення дисципліни є отримання теоретичних знань з основних принципів об'єктно-орієнтованого програмування та об'єктно-орієнтованого дизайну програмного забезпечення, вивчення основних абстрактних структур даних та особливостей їх використання їх під час розробки програмного забезпечення, вступ до проектування програмного забезпечення, елементи мови UML, що мають відношення до об'єктно-орієнтованого програмування, отримання практичних навичок використання об'єктно-орієнтованого підходу до проектування прикладного програмного забезпечення.

2. Передумови для вивчення дисципліни: алгоритмізація та програмування.

3. Опис навчальної дисципліни

Найменування показників	Характеристика навчальної дисципліни ²	
	денна форма навчання	заочна (дистанційна) форма навчання ³
Кількість кредитів - <u>8</u> Кількість годин - <u>240</u>	<u> </u>	

4. Очікувані результати навчання з дисципліни

Основними завданнями вивчення дисципліни Об'єктно-орієнтоване програмування є формування сукупності знань та вмінь для використання методів об'єктно-орієнтованого дизайну та принципів S.O.L.I.D для розробки прикладного програмного забезпечення автоматизованих інформаційних систем засобами обраної мови об'єктно-орієнтованого програмування.

По завершенні вивчення дисципліни студенти повинні володіти наступними *компетентностями*:

- здатність до абстрактного мислення, аналізу та синтезу.
- здатність застосовувати знання у практичних ситуаціях.
- здатність до пошуку, оброблення та аналізу інформації з різних джерел.
- здатність ідентифікувати, класифікувати та формулювати вимоги до

² Якщо дисципліна викладається декілька семестрів, то на кожний семестр за відповідною формою навчання заповнюється окремий стовпчик таблиці.

³ Якщо дисципліна на заочній (дистанційній) формі навчання не викладається, то графа “заочна форма навчання” відсутня.

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.

- володіння знаннями про інформаційні моделі даних, здатність створювати програмне забезпечення для зберігання, видобування та опрацювання даних.

- здатність застосовувати фундаментальні і міждисциплінарні знання для успішного розв’язання завдань інженерії програмного забезпечення.

- здатність до алгоритмічного та логічного мислення.

Результати навчання:

- 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.

- уміння вибирати та використовувати відповідну задачі методологію створення програмного забезпечення.

- вибирати вихідні дані для проектування, керуючись формальними методами опису вимог та моделювання.

- знати і застосовувати методи розробки алгоритмів, конструювання програмного забезпечення та структур даних і знань.

- уміння документувати та презентувати результати розробки програмного забезпечення.

5. Критерії оцінювання результатів навчання Підсумковий контроль знань та компетентностей студентів з навчальної дисципліни здійснюється на підставі екзамену.

Відповідність підсумкової семестрової рейтингової оцінки в балах оцінці за національною шкалою та шкалою ECTS:

Оцінка в балах	Оцінка за національною шкалою	Оцінка за шкалою ECTS	
		Оцінка	Пояснення
90-100	Відмінно	A	Відмінно (відмінне виконання лише з незначною кількістю помилок)
82 – 89	Добре	B	Дуже добре (вище середнього рівня з кількома помилками)
75 – 81		C	Добре (в загальному вірне виконання з певною кількістю суттєвих помилок)
67 – 74	Задовільно	D	Задовільно (непогано, але зі значною кількістю недоліків)
60 – 66		E	Достатньо (виконання задовольняє мінімальним критеріям)
35 – 59	Незадовільно	FX	Незадовільно (з можливістю повторного складання)
1 – 34		F	Незадовільно (з обов’язковим повторним курсом)

6. Засоби діагностики результатів навчання - тестові завдання.

7. Розподіл дисципліни у годинах за формами організації освітнього процесу та видами навчальних занять⁴

Назва теми лекційного матеріалу	Кількість годин		Назва тем	Кількість годин		Література
	очна	заочна		очна	заочна	
1	2	3	4	5	6	7
Семестр 1.						
Тема 1. Основні принципи об'єктно-орієнтованого програмування та дизайну (SOLID)	2		ПР1. Створення простого класу, робота із атрибутами класу. СРС. Основні принципи об'єктно-орієнтованого програмування та дизайну, основні принципи SOLID.	4 8		О. 1-8 Д. 1-5 ІР. 1-6
Тема 2. Реалізація відносин між об'єктами і класами. Агрегація та композиція	2		ПР2. Агрегація, композиція СРС. Діаграми класів, взаємовідносини між класами	4 8		О. 1-8 Д. 1-5 ІР. 1-6
Тема 3. Реалізація відносин між об'єктами і класами. Спадкування.	2		ПР3. Спадкування СРС. Спадкування, проблеми великих ієрархій	4 6		О. 1-8 Д. 1-5 ІР. 4-6
Тема 4. Реалізація відносин між об'єктами і класами. Поліморфізм.	2		ПР4. Поліморфізм СРС. Поліморфізм	4 6		О. 1-8 ІР. 4-6
Тема 5. Реалізація відносин між об'єктами і класами. Спадкування множинне.	2		ПР5. Множинне спадкування СРС. Спадкування множинне.	4 6		О. 1-8 ІР. 7-9
Тема 6. Перевантаження операторів.	2		ПР6. Перевантаження операторів СРС. Перевантаження операторів як засіб спрощення написання та читання програмного забезпечення.	4 6		О. 1-8 ІР. 9-14
Тема 7. Обробка виняткових ситуацій	2		ПР7. Обробка виняткових ситуацій СРС. Виняткові ситуації	4 6		О. 1-11 Д. 1-9 ІР. 18-19
Тема 8. Метакласи, елементи функційного програмування	2		ПР8. Метакласи, елементи функціонального програмування. Лямбда функції СРС. Метакласи, метапрограмування та інші технології.	4 6		О. 1-8 Д. 1-5 ІР. 20-22

⁴ Якщо дисципліна викладається декілька семестрів, то теми розбивати посеместрово.

Тема 9. Абстрактні типи даних із використанням ООП	–		ПР9. Реалізація абстрактних структур даних. Зв'язані списки СРС. Зв'язані списки	4 6		О. 1-8 Д. 1-5
			ПР10. Реалізація абстрактних структур даних. Дерева. СРС. Дерева.	4 6		
			ПР11. Реалізація абстрактних структур даних. Черги. СРС. Черги.	4 6		
			ПР12. Реалізація абстрактних структур даних. Стек. СРС. Стек.	4 6		
Тема 10. Використання ООП та ООД при розробці програм	–		ПР13. Шаблони проектування. Патерни, що породжують СРС. Патерни, що породжують	4 6		
			ПР14. Шаблони проектування. Структурні патерни СРС. Структурні патерни	4 6		
			ПР15. Шаблони проектування. Поведінкові патерни СРС. Поведінкові патерни	4 6		
			ПР16. Шаблони проектування. Поведінкові патерни. СРС. Поведінкові патерни	4 6		
Усього за семестр	16			164		
Курсова робота				30		
Іспит				30		
УСЬОГО за дисципліну	16			224		

8. Орієнтовна тематика індивідуальних та/або групових занять курсова робота на тему: «Використання об'єктно-орієнтованого програмування для розв'язання прикладної задачі», вихідні дані згідно варіанту

9. Форми поточного та підсумкового контролю усне та письмове опитування, захист практичних робіт, тестові завдання, захист курсової роботи, екзамен.

10. Інструменти, обладнання та програмне забезпечення Debian GNU Linux, ОС Windows, Python 3, Microsoft Visual Studio, Visual Studio Code.

11. Рекомендовані джерела інформації

1. Базова література

1. Объектно-ориентированный анализ и проектирование с примерами приложений, 3-е изд.: Пер. с англ. / Буч Г. и др - М.: ООО "И.Д. Вильямс", 2008. - 720 с.
2. Лутц М. Изучаем Python, том 1, 5-е изд.: Пер. с англ. — СПб.: ООО "Диалектика", 2019. — 832 с.

3. Лутц М. Изучаем Python, том 2, 5-е изд. : Пер. с англ. — СПб. : ООО “Диалектика”, 2020. — 720 с.
4. Хеллман Д. Стандартная библиотека Python 3: справочник с примерами, 2-е изд. : Пер. с англ. — СПб. : ООО “Диалектика”, 2019. — 1376 с.
5. Хайнеман Д., Поллис Г., Сеяков С. Алгоритмы. Справочник с примерами на C, C++, Java и Python, 2-е изд.: Пер. с англ. — СПб.: ООО “Альфа-книга”, 2017. — 432 с.
6. Приемы объектно-ориентированного программирования. Паттерны проектирования / [Э. Гамма, Р. Хелм, Р. Джонсон, Д. Влссидес] . – СПб: Питер, 2010. – 368 с.
7. Рамальо Л. Python. К вершинам мастерства. пер. с англ. Слинкин А. А. – М.: ДМК-Пресс, 2016. –768 с.
8. Мнушка, О.В. Об’єктно-орієнтоване програмування» : конспект лекцій для студентів за спеціальністю 121 «Інженерія програмного забезпечення» та 122 «Комп’ютерні науки» - Харків, ХНАДУ, 2020.
9. Мнушка, О.В. Методичні вказівки для практичних робіт з дисципліни «Об’єктно-орієнтоване програмування» для студентів за спеціальністю 121 «Інженерія програмного забезпечення» та 122 «Комп’ютерні науки» - Харків, ХНАДУ, 2020.
10. Мнушка, О.В. Методичні вказівки для самостійної роботи з дисципліни «Об’єктно-орієнтоване програмування» для студентів за спеціальністю 121 «Інженерія програмного забезпечення» та 122 «Комп’ютерні науки» - Харків, ХНАДУ, 2020.
11. Мнушка, О.В., Методичні вказівки по виконанню курсової роботи з дисципліни «Об’єктно-орієнтоване програмування» для студентів спеціальностей 121 «Інженерія програмного забезпечення» та 122 «Комп’ютерні науки» галузі знань 12 «Інформаційні технології» – Харків, ХНАДУ, 2020. – 20 с.

2. Допоміжна література

1. Кормен Т. Х. Алгоритмы: вводный курс. / Т.Х. Кормен : Пер. с англ. – М. : ООО “И.Д. Вильямс”, 2014. – 208 с.: ил.
2. Дейтел П., Дейтел Х. Python: Искусственный интеллект, большие данные и облачные вычисления. — СПб.: Питер, 2020. — 864 с.
3. Бусигін Б.С. Прикладна інформатика: Підручник / Б.С. Бусигін , Г.М. Коротенко, Л.М. Коротенко. – Дніпропетровськ: Національний гірничий університет, 2004. – 559 с.
4. Седер Н. Python. Экспресс-курс. 3-е изд. — СПб.: Питер, 2019. — 480 с.
5. Прохоренок Н. А. Python 3 и PyQt 5. Разработка приложений. — 2-е изд., перераб. и доп. /
6. Н. А. Прохоренок, В. А. Дронов. — СПб.: БХВ-Петербург, 2018. — 832 с.
7. Мартелли А., Рейвенскрофт А., Холден С. Python. Справочник. Полное описание языка, 3-е издание. СПб.: ООО "Диалектика", 2019. - 896 с.

3. Інформаційні ресурси

1. Welcome to Python.org.–URL: <https://www.python.org>.
2. Data model. URL: <https://docs.python.org/3.8/reference/datamodel.html>
3. S.O.L.I.D principles: what are they and why projects should use them. URL: https://medium.com/@mari_azevedo/s-o-l-i-d-principles-what-are-they-and-why-projects-should-use-them-50b85e4aa8b6
4. The clean code blog. URL: <https://blog.cleancoder.com/>
5. SOLID. URL: <https://habr.com/ru/post/348286/>
6. Association vs. Dependency vs. Aggregation vs. Composition. URL: <https://nirajrules.wordpress.com/2011/07/15/association-vs-dependency-vs-aggregation-vs-composition/>
7. Композиция или наследование: как выбрать? URL: <https://habr.com/ru/post/325478/>
8. Об'єктно-орієнтоване програмування.URL: <http://programming.in.ua/programming/basisprogramming/25-oop.html>
9. Утиная типизация в Python-3 примера. URL: <https://nuancesprog.ru/p/9142/>
- 10.Duck typing. URL: <https://devopedia.org/duck-typing>
- 11.Ликбез по типизации в языках программирования. URL: <https://habr.com/ru/post/161205/>
- 12.The Python 2.3 Method Resolution Order. URL: <https://www.python.org/download/releases/2.3/mro/>
- 13.Порядок разрешения методов в Python. URL: <https://habr.com/ru/post/62203/>
- 14.What is a Mixin. URL: <https://chrisbartos.com/articles/what-is-a-mixin/>
- 15.PEP-318. Decorators for Functions and Methods. URL: <https://www.python.org/dev/peps/pep-0318/>
- 16.Понимаем декораторы в Python'е, шаг за шагом. Шаг 1. URL: <https://habr.com/ru/post/141411/>
- 17.Понимаем декораторы в Python'е, шаг за шагом. Шаг 2. URL: <https://habr.com/en/post/141501/>
- 18.Мир Python: исключения . URL: [https://ru.hexlet.io/courses/advanced_python/lessons/python_exceptions/theory_u
nit](https://ru.hexlet.io/courses/advanced_python/lessons/python_exceptions/theory_unit)
- 19.Исключения. URL: <https://wombat.org.ua/AByteOfPython/exceptions.html>
- 20.Python Metaclasses. URL: <https://realpython.com/python-metaclasses/>
- 21.Продвинутый Python, часть 3: классы и метаклассы. URL: <https://ru.hexlet.io/blog/posts/prodvinutyy-python-chast-3-klassy-i-metaklassy>
- 22.Метаклассы и метапрограммирование в Python. URL: <https://gitjournal.tech/metaklassy-i-metaprogrammirovanie-v-python/>

Розроблено та внесено: кафедрою комп'ютерних технологій і мехатроніки
(повне найменування кафедри)

Розробник (и) програми: старший викладач Мнушка Оксана Василівна
(підпис) (ПІБ розробників)

Обговорено та рекомендовано до затвердження на засіданні кафедри
Протокол № 20 від "28" серпня 2020 р.
(номер) (та дата протоколу)

Завідувач кафедри д.т.н., проф. Ніконов Олег Якович
(науковий ступінь, вчене звання) (підпис) (ПІБ завідувача кафедри)

Погоджено

Декан Механічного факультету
(повна назва факультету, де читається дисципліна)
д.т.н., проф. Кириченко Ігор Георгійович
(науковий ступінь, вчене звання) (підпис) (ПІБ декана)
" 31 " 08 2020 року
(день) (місяць) (рік)

©Мнушка О.В., 2020 рік

Примітка:

Робоча програма навчальної дисципліни розробляється відповідною кафедрою у 2-х екземплярах на 5 років і затверджується до 30 серпня: 1 екземпляр - у навчальний відділ; 2 екземпляр залишається на кафедрі.

Форма є редакції ХНАДУ відповідно до листа МОН України за №1/9-434 від 09 липня 2018 року затверджена Методичною радою ХНАДУ 26 вересня 2018 року протокол №1