



ПРОГРАМУВАННЯ.ЧАСТИНА-2.

Об'єктно-орієнтоване програмування Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	<i>Перший (бакалаврський)</i>
Галузь знань	<i>12 Інформаційні технології</i>
Спеціальність	<i>123 Комп'ютерна інженерія</i>
Освітня програма	<i>Комп'ютерні системи та мережі</i>
Статус дисципліни	<i>Нормативна</i>
Форма навчання	<i>очна(денна)/заочна</i>
Рік підготовки, семестр	<i>1 курс, весняний семестр</i>
Обсяг дисципліни	<i>5 кредитів ЕКТС/150 годин</i>
Семестровий контроль/ контрольні заходи	<i>Залік</i>
Розклад занять	a0e5-58866f4b3e19
Мова викладання	<i>Українська</i>
Інформація про керівника курсу / викладачів	Лектор та лабораторні: <i>Алещенко Олексій Вадимович (старший викладач), alexey.aleshchenko@gmail.com, +38-096-833-89-99 (години для комунікації: 09:00-21:00)</i>
Розміщення курсу	https://asdonok.github.io/OOP-SITE/

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Метою навчальної дисципліни є вивчення методів та засобів програмування задач довільної складності з використанням методології, що базується на об'єктно-орієнтованій парадигмі. За результатами вивчення дисципліни студент має бути здатним вирішувати професійні завдання та володіти такими компетенціями: сформулювати задачу для вирішення на визначеному типі комп'ютерної системи або визначити чинники, за якими розв'язок цієї задачі на цьому типі системи є недоцільним чи неможливим; визначити, чи є оптимальною для розв'язку конкретної задачі технологія об'єктно-орієнтованого програмування та обрати мову, систему програмування та інструментальне середовище; розробити структуру класів, необхідну для розв'язку задачі відповідно до об'єктно-орієнтованої технології програмування; виконати розробку коду заданих програми; виправити синтаксичні та семантичні помилки та рефакторинг коду (налагодити програму); оформити документацію, що має супроводжувати програмний продукт у відповідності до державних чи міжнародних стандартів; читати й аналізувати тексти готових програм.

Дисципліна забезпечує наступні програмні результати навчання освітньо-професійної програми Інженерія програмного забезпечення комп'ютерних систем: ПРН2,ПРН5-7, ПРН9-10, ПРН15, ПРН22-24.

Предметом навчальної дисципліни є програмні засоби обчислювальної техніки що необхідні для виконання майбутніми фахівцями професійних обов'язків у напрямку створення програмного забезпечення для комп'ютерних систем (КС) в умовах колективу з використанням сучасних мов та бібліотек об'єктно-орієнтованого програмування з урахуванням особливостей майбутньої професії і можливих первинних посад бакалаврів.

Згідно з вимогами освітньо-професійної програми студенти після засвоєння навчальної дисципліни мають продемонструвати такі **результати навчання**:

- Здатність до абстрактного мислення, аналізу та синтезу;
- Вміння виявляти, ставити та вирішувати проблеми;
- Здатність працювати в команді;
- Здатність використовувати сучасні методи і мови програмування для розроблення алгоритмічного та програмного забезпечення;
- Здатність створювати системне та прикладне програмне забезпечення комп'ютерних систем та мереж;
- Здатність забезпечувати захист інформації, що обробляється в комп'ютерних та кіберфізичних системах та мережах з метою реалізації встановленої політики інформаційної безпеки;
- Готовність брати участь у роботах з впровадження комп'ютерних систем та мереж, введення їх до експлуатації на об'єктах різного призначення;
- Здатність системно адмініструвати, використовувати, адаптувати та експлуатувати наявні інформаційні технології та системи;
- Здатність ідентифікувати, класифікувати та описувати роботу програмно-технічних засобів, комп'ютерних та кіберфізичних систем, мереж та їхніх компонентів шляхом використання аналітичних методів і методів моделювання;
- Здатність проектувати, впроваджувати, адмініструвати та обслуговувати глобальні, локальні інтелектуальні програмно - конфігуровані комп'ютерні мережі;
- Здатність розробляти, адаптувати, використати програмне забезпечення для покращення ефективності застосування високопродуктивних комп'ютерних систем;
- Здатність організації обчислювальних процесів в високопродуктивних комп'ютерних системах з різною структурною організацією на основі використання новітніх технологій планування і диспетчеризації та сучасних операційних систем.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

Матеріал курсу пов'язаний з матеріалами, які вивчаються у навчальних дисциплінах «Комп'ютерна логіка», «Алгоритми та структури даних», «Вища математика», «Програмування-1. Програмування».

Матеріал даної навчальної дисципліни буде застосований при вивченні наступних навчальних дисциплін: «Дискретна математика», «Алгоритми та методи обчислень», «Комп'ютерне моделювання», «Програмна інженерія», «Системне програмування», «Технологія розподілених обчислень», «Мережні та інформаційні технології», «Комп'ютерне моделювання», «Основи програмної інженерії», «Системне програмування», «Технологія розподілених обчислень», «Мережеві та інформаційні технології» та інші.

3. Зміст навчальної дисципліни

Тема 1.1 Класи та об'єкти. Наслідування, інкапсуляція та поліморфізм.

Тема 1.2 Загальна характеристика мови Java. Порівняння з іншими мовами програмування. Класи та методи в Java. Пакети та інтерфейси. Прості типи та основи синтаксису Java.

Тема 1.3 Робота з масивами в Java.

Тема 1.4 Робота зі строками в Java. Класи String, String Buffer, String Builder.

Тема 1.5 Перевантаження методів. Перевантаження конструкторів. Управління доступом. Статичні елементи.

Тема 1.6 Властивості (поля). Наслідування властивостей. Управління областю видимості. Внутрішні та анонімні класи.

Тема 1.7 Створення багаторівневих ієрархій. Клас Object.

Тема 1.8 Методи. Наслідування методів. Поліморфізм. Специфікатори super та final.

Тема 1.9 Реалізація поліморфізму. Перевизначення методів. Раннє та пізнє зв'язування.

Тема 1.10 Колекції в Java. Зв'язані списки.

Тема 1.11 Обробка виключень та модульне тестування.

Тема 1.12 Управління пам'яттю, деструктори об'єктів та Garbage Collector.

Тема 1.13 Пакет java.util.stream.

Тема 1.14 Події та орієнтоване на події управління поведінкою програми.

4. Навчальні матеріали та ресурси

Базова:

1. Алещенко О.В. Програмування-2. Об'єктно-орієнтоване програмування: Методичні вказівки до виконання лабораторних робіт. [Електронний ресурс]. – Київ : Кафедра ОТ, НТУУ "КПІ", 2021. Режим доступу: <https://cloud.comsys.kpi.ua/s/ExxaNc48WjinAqY>
2. Алещенко О.В. Програмування-2. Об'єктно-орієнтоване програмування: конспект лекцій. [Електронний ресурс]. – Київ : Кафедра ОТ, НТУУ "КПІ", 2021. Режим доступу: https://drive.google.com/drive/folders/1_09ZizH-0YBZ1z13HX5j9giRQg42EC5Y?usp=sharing

Додаткова:

1. Бублик, Володимир Васильович. Об'єктно-орієнтоване програмування : підручник для студентів, які навчаються за напрямками "Комп'ютерні науки", "Комп'ютерна інженерія", "Програмна інженерія", "Системна інженерія", "Інформатика", "Прикладна математика" / В.В. Бублик. - Київ : ІТ-книги, 2015. - 637 с.
2. Алхімова С. М. Об'єктно-орієнтоване програмування. Об'єктно-орієнтоване програмування мовою високого рівня С++ : методичні вказівки до виконання лабораторних. - Київ : НТУУ "КПІ", 2015. - 48 с.
3. Алхімова С. М. Алгоритмізація та програмування. Програмування мовою високого рівня С++ : методичні вказівки до виконання лабораторних робіт. - Київ : НТУУ "КПІ", 2015. - 105 с.
4. Крячок О. С., Кузьменко І. М., Гурін А. Л. Основи програмування та алгоритмічні мови - 2. Спеціальні засоби мови програмування : методичні вказівки до виконання комп'ютерних практикумів. - Київ : НТУУ "КПІ", 2012.
5. Дичка І. А., Зорін Ю. М., Орлова М. М. Алгоритми та методи обчислень : метод. вказівки до викон. лаборатор. робіт. - Київ : НТУУ "КПІ", 2011. - 64 с.
6. K.Wiegers Creating a software engineering culture // Dorset House Publ.-New York, 2003.- 358 p.

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

Очна форма навчання

Назви розділів і тем	Кількість годин			
	Всього	у тому числі		
		Лекції	Практичні (семінарські)	Лабораторні (комп'ютерні й практикум)
1	2	3	4	5

Тема 1.1 Класи та об'єкти. Наслідування, інкапсуляція та поліморфізм.		1		
Тема 1.2 Загальна характеристика мови Java. Порівняння з іншими мовами програмування. Класи та методи в Java. Пакети та інтерфейси. Прості типи та основи синтаксису Java.		3		3
Тема 1.3 Робота з масивами в Java.		2		3
Тема 1.4 Робота зі строками в Java. Класи String, String Buffer, String Builder.		2		3
Тема 1.5 Перевантаження методів. Перевантаження конструкторів. Управління доступом. Статичні елементи.		2		
Тема 1.6 Властивості (поля). Наслідування властивостей. Управління областю видимості. Внутрішні та анонімні класи.		4		3
Тема 1.7 Створення багаторівневих ієрархій. Клас Object.		2		3
Тема 1.8 Методи. Наслідування методів. Поліморфізм. Специфікатори super та final.		2		3
Тема 1.9 Реалізація поліморфізму. Перевизначення методів. Раннє та пізнє зв'язування.		2		
Тема 1.10 Колекції в Java. Зв'язані списки.		6		
Тема 1.11 Обробка виключень та модульне тестування		4		
Тема 1.12 Управління пам'яттю, деструктори об'єктів та Garbage Collector.		2		
Тема 1.13 Пакет java.util.stream		2		
Тема 1.14 Події та орієнтоване на події управління поведінкою програми.		2		
Всього в семестрі:	54	36		18

Назви розділів і тем	Кількість годин			
	Всього	у тому числі		
		Лекції	Практичні (семінарські)	Лабораторні (комп'ютерний практикум)
1	2	3	4	5
Тема 1.1 Класи та об'єкти. Наслідування, інкапсуляція та поліморфізм.		0,5		
Тема 1.2 Загальна характеристика мови Java. Порівняння з іншими мовами програмування. Класи та методи в Java. Пакети та інтерфейси. Прості типи та основи синтаксису Java.		0,5		1
Тема 1.3 Робота з масивами в Java.		0,5		1
Тема 1.4 Робота зі строками в Java. Класи String, String Buffer, String Builder.		0,5		1
Тема 1.5 Перевантаження методів. Перевантаження конструкторів. Управління доступом. Статичні елементи.		1		
Тема 1.6 Властивості (поля). Наслідування властивостей. Управління областю видимості. Внутрішні та анонімні класи.		0,5		1
Тема 1.7 Створення багаторівневих ієрархій. Клас Object.		0,5		1
Тема 1.8 Методи. Наслідування методів. Поліморфізм. Специфікатори super та final.		0,5		1
Тема 1.9 Реалізація поліморфізму. Перевизначення методів. Раннє та пізнє зв'язування.		0,5		
Тема 1.10 Колекції в Java. Зв'язані списки.		0,5		
Тема 1.11 Обробка виключень та модульне тестування		0,5		
Тема 1.12 Управління пам'яттю, деструктори об'єктів та Garbage Collector.		0,5		
Тема 1.13 Пакет java.util.stream		0,5		
Тема 1.14 Події та орієнтоване на події управління поведінкою програми.		0,5		
Всього в семестрі:	14	8		6

№ лекції	Назва теми лекції, перелік основних питань
1	«Вступ в ООП на мові програмування Java». Класи та об'єкти. Наслідування, інкапсуляція та поліморфізм. Загальна характеристика мови Java. Порівняння з іншими мовами програмування. Прості типи та основи синтаксису Java.
2	«Вступ в ООП на мові програмування Java. Продовження». Класи та методи в Java. Пакети та інтерфейси.
3	«Робота з масивами в Java». Одновимірні, багатовимірні масиви та об'єкти, що представляють масиви в Java.
4	«Робота зі строками в Java». Класи String, String Buffer, String Builder.
5	«Методи класу. Перевантаження, статичність та доступ». Перевантаження методів. Перевантаження конструкторів. Управління доступом. Статичні елементи.
6	«Анонімні класи на прикладі сортування масиву об'єктів». Анонімні класи. Інтерфейс Comparator. Метод Compare. Сортування масиву об'єктів. Клас Arrays. Метод Sort.
7	«Властивості (поля) класу. Наслідування та управління їх видимістю». Властивості (поля). Наслідування властивостей. Управління областю видимості.
8	«Ієрархії об'єктів». Створення багаторівневих ієрархій об'єктів через наслідування та/або агрегацію. Клас Object.
9	«Наслідування методів. Поліморфізм». Наслідування методів. Поліморфізм. Специфікатори super та final.
10	«Реалізація поліморфізму». Реалізація поліморфізму. Перевизначення методів. Ранне та пізнє зв'язування.
11	«Колекції в Java». Колекції в Java. Зв'язані списки. Інтерфейси Set та List.
12	«Колекції в Java. Продовження». Методи toArray. Ітератори.
13	«Колекції в Java. Продовження». Унікальність елементів у колекціях типу Set. Еквівалентність об'єктів.
14	«Обробка виключень». Класи опису виключних ситуацій. Власні виключні ситуації. Exception, Error, Runtime Exception.
15	«Модульне тестування». Використання модулів JUnit. Клас Assert. Анотації Before, After. Методи AssertEquals, AssertNull, AssertTrue, AssertFalse.
16	«Управління пам'яттю, деструктори об'єктів та Garbage Collector». Система «збору сміття». Метод finalize.
17	«Пакет java.util.stream». Функціональний стиль програмування. Чисті функції. Технологія map-reduce.
18	«Події та орієнтоване на події управління поведінкою програми». Події в Java. Модулі AWT та Swing. Observer, JMS.

Очна форма навчання

№ з/п	Назва лабораторної роботи	Кількість ауд. годин
1	Основні типи та оператори мови програмування Java (Тема 1.2)	3
2	Масиви в мові програмування Java (Тема 1.3)	3
3	Рядки в мові програмування Java (Тема 1.4)	3
4	Класи в мові програмування Java (Тема 1.6)	3
5	Відношення між класами в мові програмування Java (Тема 1.7)	3

6	Наслідування та поліморфізм (Тема 1.8)	3
7	Робота з колекціями в мові програмування Java (Тема 1.10)	-
8	Обробка виключних ситуацій та основи тестування в мові програмування Java (Тема 1.11)	-
18	«Події та орієнтоване на події управління поведінкою програми». Події в Java. Модулі AWT та Swing. Observer, JMS.	
	Разом	18

Заочна форма навчання

№ з/п	Назва лабораторної роботи	Кількість ауд. годин
1	Основні типи та оператори мови програмування Java (Тема 1.2)	1
2	Масиви в мові програмування Java (Тема 1.3)	1
3	Рядки в мові програмування Java (Тема 1.4)	1
4	Класи в мові програмування Java (Тема 1.6)	1
5	Відношення між класами в мові програмування Java (Тема 1.7)	1
6	Наслідування та поліморфізм (Тема 1.8)	1
7	Робота з колекціями в мові програмування Java (Тема 1.10)	-
8	Обробка виключних ситуацій та основи тестування в мові програмування Java (Тема 1.11)	-
	Разом	6

6. Самостійна робота студента

№ лекції	Назва теми лекції, завдання на СРС
1	«Вступ в ООП на мові програмування Java». Частково або повністю виконати першу лабораторну роботу даного кредитного модуля на мові програмування Java.
2	«Вступ в ООП на мові програмування Java. Продовження». Створити декілька методів в декількох різних класах, які знаходяться в різних пакетах та реалізують різні інтерфейси.
3	«Робота з масивами в Java». Частково або повністю виконати другу лабораторну роботу даного кредитного модуля на мові програмування Java.
4	«Робота зі строками в Java». Частково або повністю виконати третю лабораторну роботу даного кредитного модуля на мові програмування Java.
5	«Методи класу. Створити декілька перевантажених методів та декілька конструкторів в класі, які мають різний рівень доступу та різну статичність.
6	«Анонімні класи на прикладі сортування масиву об'єктів». Частково або повністю виконати четверту лабораторну роботу даного кредитного модуля на мові програмування Java.
7	«Властивості (поля) класу. Наслідування та управління їх видимістю». Створити клас з декількома полями з різними областями видимості. Створити клас, що наслідує попередній. Спробувати доступитися в ньому до полів попереднього класу та створити власні поля з такими ж іменами.
8	«Ієрархії об'єктів». Частково або повністю виконати п'яту лабораторну роботу даного кредитного модуля на мові програмування Java.
9	«Наслідування методів. Поліморфізм». Частково або повністю виконати шосту лабораторну роботу даного кредитного модуля на мові програмування Java.

10	«Реалізація поліморфізму». Створити клас з декількома методами. Створити клас, що наслідуює попередній та перевизначає його методи. Створити екземпляр другого класу, посилання на який буде зберігати змінна першого класу. Спробувати доступитися через вказану змінну першого класу до перевизначеного методу з другого класу.
11	«Колекції в Java». Частково або повністю виконати сьому лабораторну роботу даного кредитного модуля на мові програмування Java.
12	«Колекції в Java. Продовження». Частково або повністю виконати сьому лабораторну роботу даного кредитного модуля на мові програмування Java.
13	«Колекції в Java. Продовження». Частково або повністю виконати сьому лабораторну роботу даного кредитного модуля на мові програмування Java.
14	«Обробка виключень». Виконати частину восьмої лабораторної роботи даного кредитного модуля на мові програмування Java, яка відноситься до обробки виключних ситуацій.
15	«Модульне тестування». Виконати частину восьмої лабораторної роботи даного кредитного модуля на мові програмування Java, яка відноситься до модульного тестування.
16	«Управління пам'яттю, деструктори об'єктів та Garbage Collector». Ознайомитися з алгоритмом роботи Garbage Collector в Java.
17	«Пакет java.util.stream». Реалізувати алгоритм виборки з масиву за критерієм за допомогою пакету java.util.stream.
18	«Події та орієнтоване на події управління поведінкою програми». Реалізувати програму на мові програмування Java, яка використовує механізм подій.

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

- правила захисту лабораторних робіт:
 - студент надає програмний код до лабораторної роботи,
 - студент захищає наданий програмний код шляхом співбесіди;
- правила захисту індивідуальних завдань:
 - студент надає програмний код до індивідуального завдання,
 - студент захищає наданий програмний код шляхом співбесіди;
- політика дедлайнів та перескладань:
 - студент має право здавати та перездавати лабораторні роботи до дня заліку включно,
 - обмеження на кількість перескладань відсутнє;
- політика щодо академічної доброчесності:
 - студент має право захищати програмний код, який був написаний не лише ним, але в цьому випадку студент має розуміти «що» та «як» виконує цей код, вміти використати його як основу для внесення модифікацій, які можуть бути запропоновані викладачем.

8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Рейтинг студента на курсі складається з балів, які він/вона отримує за захищені лабораторні роботи (R1) та заліковий семестровий контроль (R2).

$$R_s = R_1 + R_2 = 100 \text{ балів}$$

В результаті максимальний середній бал дорівнює:

- ((6 лабораторних робіт x 15 балів) або індивідуальне завдання) = 90 балів,
- заліковий семестровий контроль = 10 балів.

Календарний контроль: провадиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу. Перший календарний контроль ставиться по факту виконання перших двох робіт, а другий - ще двох робіт додатково.

Семестровий контроль: залік - проводиться у вигляді співбесіди та/або підведення підсумків за результатами в семестрі.

Умови допуску до семестрового контролю: мінімально позитивна оцінка за індивідуальне завдання / зарахування усіх лабораторних робіт / семестровий рейтинг більше 30 балів.

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

<i>Кількість балів</i>	<i>Оцінка</i>
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

9. Додаткова інформація з дисципліни (освітнього компонента)

Є можливість зарахування сертифікатів проходження дистанційних чи онлайн курсів за тематикою дисципліни. Побажання: спробуйте взяти для себе якомога більше корисного та цікавого.

Робочу програму навчальної дисципліни (силабус):

Складено старшим викладачем кафедри обчислювальної техніки Алещенком Олексієм Вадимовичем

Ухвалено кафедрою обчислювальної техніки (протокол № 10 від 25.05.2022)

Погоджено Методичною комісією ФІОТ (протокол № 10 від 9.06.2022)

.....