

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЗАПОРІЗЬКА ПОЛІТЕХНІКА»

Кафедра системного аналізу та обчислювальної математики
(найменування кафедри)

СИЛАБУС НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Об'єктно-орієнтований підхід в задачах системного аналізу
(назва навчальної дисципліни)

Освітня програма: інтелектуальні технології та прийняття рішень в складних системах
(назва освітньої програми)

Спеціальність: 124 Системний аналіз
(найменування спеціальності)

Галузь знань: 12 Інформаційні технології
(найменування галузі знань)

Ступінь вищої освіти: бакалаврський
(назва ступеня вищої освіти)

Затверджено на засіданні кафедри

СА та ОМ
(найменування кафедри)

Протокол № 18 від 16.08.2021 р.

м. Запоріжжя 2021

1. Загальна інформація	
Назва дисципліни	<i>Об'єктно-орієнтований підхід в задачах системного аналізу</i>
Рівень вищої освіти	<i>Перший (бакалаврський) рівень</i>
Викладач	<i>Широкорад Дмитро Вікторович, к.ф.-м.н., ст. викладач</i>
Контактна інформація викладача	<i>Телефон кафедри 769-8247, Телефон викладача +380631182567, hoveringphoenix@gmail.com</i>
Час і місце проведення навчальної дисципліни	<i>Предметна аудиторія кафедри, комп'ютерні класи згідно розкладу занять</i>
Обсяг дисципліни	<i>розподіл годин: лекції-14, лабораторні заняття-14, вид контролю- залік,</i>
Консультації	<i>Згідно з графіком консультацій</i>
2. Пререквізити і постреквізити навчальної дисципліни	
<i>Пререквізити: основи математичного аналізу, лінійна алгебра, програмування та алгоритмічні мови</i>	
3. Характеристика навчальної дисципліни	
Загальні компетентності: – K01. Здатність до абстрактного мислення, аналізу та синтезу; – K02. Здатність застосовувати знання у практичних ситуаціях – K04. Знання та розуміння предметної області та розуміння професійної діяльності; – K07. Здатність до пошуку, оброблення та аналізу інформації з різних джерел; – K10. Здатність працювати автономно; – K12. Здатність працювати в команді. Спеціальні компетентності: – K18. Здатність формалізувати проблеми, описані природною мовою, у тому числі за допомогою математичних методів, застосовувати загальні підходи до математичного моделювання конкретних процесів; – K22. Здатність до комп'ютерної реалізації математичних моделей реальних систем і процесів; проектувати, застосовувати і супроводжувати програмні засоби моделювання, прийняття рішень, оптимізації, обробки інформації, інтелектуального аналізу даних. Програмні результати навчання: – ПР08. Володіти сучасними методами розробки програм і програмних комплексів та прийняття оптимальних рішень щодо складу програмного забезпечення, алгоритмів процедур і операцій. – ПР09. Вміти створювати ефективні алгоритми для обчислювальних задач системного аналізу та систем підтримки прийняття рішень. В результаті вивчення дисципліни «Об'єктно-орієнтований підхід в задачах системного аналізу» студент повинен знати: – основні ідеї та поняття сучасного програмування, об'єктно-орієнтованого підходу при розробці складних програм; – сучасні методи аналізу, проектування та програмування складних програмних систем; – об'єктно-орієнтовану мову програмування C++, – характеристики інших мов програмування, їхні можливості та обмеження; – принципи побудови новітніх візуальних середовищ для розробки програм із застосуванням візуального програмування; – методи та засоби створення програм, що діють у інтерфейсному середовищі таких операційних систем, як Windows та UNIX. вміти: – виконати програмну реалізацію наданого алгоритму, проектування, розробку чи доробку програмного продукту проблемного або системного характеру із застосуванням ідей сучасного програмування, скласти програму мовою C/C++; – створити інтерфейс програми із застосуванням засобів візуального програмування; – створити бібліотечні засоби і проблемне розширення мови програмування;	

– оволодіти у разі необхідності іншою мовою програмування, виконати вибір мови програмування для розв’язання проблеми.			
4. Мета вивчення навчальної дисципліни			
Метою викладання дисципліни "Об'єктно-орієнтований підхід в задачах системного аналізу" є отримання студентом базових знань та практичних навичок з основ сучасної технології створення складних програмних продуктів на базі ідей і принципів об'єктно-орієнтованого методу, візуального програмування. Такі знання можуть бути використані для побудови програм аналізу складних систем, а також у розробках як системного, так і прикладного програмного забезпечення ЕОМ з урахуванням сучасних вимог у відношенні до надійності, якості інтерфейсу та ефективності програмних продуктів. Отримані знання та практичні навички мають служити базою для опанування у подальшому нових майбутніх систем програмування			
5. Завдання вивчення дисципліни			
Завдання: навчити студентів застосовувати сучасні мови програмування високого рівня, а також самостійно створювати власні прикладні програми з використанням концепцій об'єктно-орієнтованого програмування.			
6. Зміст навчальної дисципліни			
У навчальній дисципліні «Об'єктно-орієнтований підхід в задачах системного аналізу» розширюються знання студентів з технологій сучасного об'єктно-орієнтованого програмування за допомогою алгоритмічних мов на прикладі мови C++. Вивчається концепція об'єкту та класу, його компонентів та методів, атрибуту доступу. Курс містить відомості про перевантаження операцій та обробку виключень. Розглядаються поняття наслідування, поліморфізму та інкапсуляції. Вивчаються методи обробки виключень та базові підходи до візуального програмування.			
7. План вивчення навчальної дисципліни			
№ тижня	Назва теми	Форми організації навчання	Кількість годин
1	Класи, компоненти об'єкту, методи класу Поняття класу, його загальна структура. Компоненти класу. Дані і функції класу. Статичні компоненти та функції класу. Класові типи та покажчики. Створення і робота функцій-компонентів класу (методів).	Лекції, лабораторні заняття, самостійна робота	2
2	Конструктори, ініціалізація та перетворення об'єктів. Деструктори Конструктори і деструктори. Поняття контейнерного та вкладеного класів. Перетворення типу для класів.	Лекції, лабораторні заняття, самостійна робота	4
3	Захист даних та атрибути доступу Захист даних. Підвищення надійності програмування: атрибути доступу private, protected, public, їхня дія, переваги та недоліки їхнього застосування. Забезпечення доступу до компонентів класу із зовнішніх функцій, оголошення та застосування дружніх функцій.	Лекції, лабораторні заняття, самостійна робота	4
4	Перевантаження операцій. Перевантаження унарних та бінарних операцій Перевантаження як форма поліморфізму. Перевантаження операції присвоєння. Динамічне присвоєння. Перевантаження операції індексації (доступу до елемента масиву) []. Перевантаження символів	Лекції, лабораторні заняття, самостійна робота	4

	операцій “<<”, “>>” для програмування операцій введення/виведення.		
5	Успадкування та ієрархії класів. Віртуальні функції Створення потоків ifstream, ofstream, зв'язування їх з файлами. Операції введення/виведення. Керування форматами. Безформатне введення/виведення. Файли з довільним доступом.	Лекції, лабораторні заняття, самостійна робота	2
6	Абстрактні класи та чисті функції Поліморфізм та абстрактні класи. Віртуальні функції у ієрархіях успадкування, їхнє призначення та побудова. Поняття чистої функції. Абстрактні класи та інтерфейс класу. Особливості абстрактних класів, їхня роль у побудові складних ієрархій класів.	Лекції, лабораторні заняття, самостійна робота	2
7	Шаблони класів та їхня реалізація. Приклади побудови шаблонів класів Шаблони класів та їхня реалізація. Побудова і використання шаблонів класів. Шаблони як абстрактні типи. Приклади побудови параметризованого класу. Клас для роботи з множинами. Спеціалізація класу з шаблонами. Спеціальне визначення шаблонного класу.	Лекції, лабораторні заняття, самостійна робота	4
8	Стандартні бібліотеки програмних компонентів. Побудова бібліотечних модулів Створення потоків ifstream, ofstream, зв'язування їх з файлами. Перевантаження поточкових операцій, створення альтернативних засобів введення/виведення для класів. Переваги і недоліки стандартних поточкових операцій введення/виведення. Побудова проблемно-орієнтованих бібліотечних модулів.	Лекції, лабораторні заняття, самостійна робота	1
9	Принципи об'єктно-орієнтованої розробки SOLID SRP: принцип єдиної відповідальності. OCP: принцип відкритості / закритості. LSP: принцип заміщення Ліскова. ISP: принцип поділу інтерфейсу. DIP: принцип інверсії залежностей.	Лекції, лабораторні заняття, самостійна робота	5

8. Самостійна робота

Кількість годин, що плануються на виконання завдань самостійної роботи, тем завдань, включно з темами курсових робіт, наведені у таблиці п. 4 та у п. 9.

Необхідним елементом успішного засвоєння навчального матеріалу дисципліни є самостійна робота студентів з вітчизняною та закордонною спеціальною літературою. Самостійна робота є основним засобом оволодіння навчальним матеріалом у час, вільний від обов'язкових навчальних занять.

Основні види самостійної роботи, запропоновані студентам:

- вивчення лекційного матеріалу;
- робота з рекомендованою літературою;
- вивчення окремих тем або питань, що передбачені для самостійного опрацювання;
- вивчення основних термінів та понять з галузі регресійного аналізу;

- підготовка до лабораторних занять;
- підготовка до проміжного та підсумкового контролю;
- контрольна перевірка кожним студентом особистих знань за питаннями для самостійного поглибленого вивчення та самоконтролю;
- робота над індивідуальним завданням.

9. Система та критерії оцінювання курсу

Проміжний контроль знань студентів включає такі заходи:
виконання та захист лабораторних робіт, що передбачені учбовим планом;
аудиторні контрольні роботи (2 за семестр), що проводяться перед кожним підсумковим контролем;

Підсумковий контроль знань студентів включає залік;

Оцінювання знань при проміжному контролі можливо таким чином:

Розподіл балів, які отримують студенти

За результатами роботи протягом семестру студент отримує:

за відмінний захист лабораторних робіт – 50 балів;

при відмінному виконанні всіх завдань підсумкової контрольної роботи (контрольної роботи для заочної форми) – 50 балів;

10. Політика курсу

Академічна доброчесність. Студенти не видають за свої результати роботи інших людей. При використанні чужих ідей і тверджень у власних роботах обов'язково посилаються на використані джерела інформації. Під час оцінювання результатів навчання не користуються недозволеними засобами, самостійно виконують навчальні завдання, завдання поточного та підсумкового контролю.

Відвідування занять. Очікується, що всі студенти відвідають усі лекції та практичні заняття курсу. За згоди декана та викладача дозволяється перейти на індивідуальний графік занять. У будь-якому випадку студенти зобов'язані дотримуватися термінів виконання усіх видів робіт, передбачених робочою програмою курсу.

В межах курсу студентам можуть бути зараховані окремі результати неформального та інформального навчання, зокрема результати навчання, здобуті за курсами, :
Програмування на мові C++ (<https://stepik.org/course/7/promo>). Відповідні результати враховуються у частині, що відповідає завданням вивчення та змісту навчальної дисципліни. За умови підтвердження сертифікатом студент отримує 60 балів за відповідні види поточного контролю. Отримання вищої оцінки можливо за результатами співбесіди з викладачем з питань, що стосуються здобутих результатів неформального та/або інформального навчання.

11. Література

1. Бублик В.В. Об'єктно-орієнтоване програмування: підручник для студентів, які навчаються за напрямками «Комп'ютерні науки», «Комп'ютерна інженерія», «Програмна інженерія», «Системна інженерія», «Інформатика», «Прикладна математика» – Київ : ІТ-книга, 2015. – 624 с.
2. Бондарев В.М. Програмування на C++. – Харків, 2005. – 284 с.
3. Lafore R. Object-Oriented Programming in C++. – Indianapolis : Sams Publishing, 2002. – 1012 с.
4. Васильєв О. Програмування C++ в прикладах і задачах. – К. : Ліра-К, 2017. – 382 с.
5. Девіс С. C++ для чайників. – К. : Діалектика, 2020. – 800 с.
6. Глушаков С.В., Коваль А.В., Смирнов С.В. Язык программирования C++: Учебный курс. – Харків: Фолио; М.: Изд. АСТ, 2001. – 500 с.
7. Прата С. C/C++. Язык программирования C++. Лекции и упражнения – К.: Діалектика, 2016. – 1244 с..
8. Б. Страуструп. Язык программирования C++. Краткий курс. – К. : «Вильямс», 2020. – 1000 с.

9. Пінчук В.П., Лозовська Л.І. Програмування мовою C/C++ з прикладами та вправами. Навч. пос. – Запоріжжя: ЗНТУ, 2009. – 204 с.

12. Публікації з тематики курсу

1. Shyrokorad, D. V. Evolution of the Ni-Al Janus-like clusters under the impacts of low-energy Ar and Ar13 projectiles / D. V. Shyrokorad, G. V. Kornich, S. G. Buga // *Materials Today Communications*. – V. 23. (2020) 101107-12 (Изд. Elsevier) – doi: 10.1016/j.mtcomm.2020.101107.
2. 18. Shyrokorad D.V. Formation of the core-shell structures from bimetallic Janus-like nanoclusters under low-energy Ar and Ar13 impacts: A molecular dynamics study / D. V. Shyrokorad, G. V. Kornich, S. G. Buga // *Computational Materials Science*. – 2019. – V. 159. – P. 110-119.
3. 17. Широкопад Д. В. Моделирование столкновительной стадии эволюции двудольных биметаллических кластеров под действием димеров аргона низких энергий / Д. В. Широкопад, Г. В. Корнич // *Металлофиз. новейшие технол.* – 2017. – Т. 39, № 2. – С. 151–163.