

ХМЕЛЬНИЦЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ

**Факультет програмування та комп'ютерних і телекомунікаційних систем
Кафедра комп'ютерної інженерії та системного програмування**

ЗАТВЕРДЖУЮ
Декан ФПКТС *Савенко О.С.*
2020 р.

СИЛАБУС

Навчальна дисципліна Об'єктно-орієнтоване програмування

Освітньо- професійна програма Автоматизація та комп'ютерно-інтегровані технології

Рівень вищої освіти перший (бакалаврський)

Загальна інформація

| Позиція | Зміст інформації |
|---------------------------|--|
| Викладач(і) | Лисенко Сергій Миколайович |
| Профайл викладача | http://ki.khnu.km.ua/team/sergii-lysenko/ |
| E-mail викладача(ів) | sprlysenko@gmail.com |
| Контактний телефон | заповнюється за домовленістю |
| Сторінка дисципліни в ІСУ | https://msn.khnu.km.ua/course/view.php?id=6468 |
| Навчальний рік | 2020-2021 |
| Консультації | Очні: середа, 6-а пара, 1-114; п'ятниця, 6-а пара, 1-114; онлайн: за необхідністю та попередньою домовленістю |

Характеристика дисципліни

| Статус дисципліни | Форма навчання | Курс | Семестр | Загальне навантаження | | Кількість годин | | | | | | | | Форма семестрового контролю | |
|-------------------|----------------|------|---------|-----------------------|--------|-------------------|--------|--------------------|-------------------|-------------------------------|-------------------------------|-----------------|----------------|-----------------------------|-------|
| | | | | Кредити ЄКТС | Години | Аудиторні заняття | | | | Індивідуальна робота студента | Самостійна робота, в т.ч. ІРС | Курсовий проект | Курсова робота | Залік | Іспит |
| | | | | | | Разом | Лекції | Лабораторні роботи | Практичні заняття | | | | | | |
| О | Д | 2 | 3 | 5.0 | 150 | 68 | 34 | 34 | | | 82 | - | - | | + |

Анотація дисципліни

Дисципліна "Об'єктно-орієнтоване програмування" відноситься до циклу дисциплін професійної підготовки, забезпечує базову підготовку студентів спеціальності "Автоматизація та комп'ютерно-інтегровані технології" з програмування та характеризується широким міждисциплінарним підходом.

Мета і завдання дисципліни

Метою курсу є оволодіння студентами основними принципами і методами об'єктно-орієнтованого програмування, які в подальшому можуть сприяти їх успішному застосуванню в професійній діяльності.

Завдання дисципліни “Об’єктно-орієнтоване програмування” навчити застосовувати принципи об’єктно-орієнтованого програмування на всіх етапах життєвого циклу прикладної програмної системи, починаючи з аналізу вимог до програмної системи і її попереднього проектування, і закінчуючи її реалізацією, тестуванням і наступним супроводом.

Пререковізити: Англійська мова; Програмування; **Кореквізити:** Веб-технології в автоматизованих системах; Програмування мікропроцесорних систем керування; Людино-машиний інтерфейс та програмування систем реального часу; Проектування багаторівневих систем керування і збору даних

Очікувані результати навчання.

Студент, який успішно завершив вивчення дисципліни, повинен: застосовувати сучасні інформаційні технології та мати навички розробляти алгоритми та комп’ютерні програми з використанням мов високого рівня та технологій об’єктно-орієнтованого програмування мовою C++, обґрунтовувати вибір структури та розробляти прикладне програмне забезпечення для систем управління на базі локальних засобів автоматизації на основі об’єктно-орієнтованого програмування мовою C++, виконувати роботи з проектування систем автоматизації, знати зміст і правила оформлення проектних матеріалів, склад проектної документації та послідовність виконання проектних робіт з врахуванням вимог відповідних нормативно-правових документів та міжнародних стандартів.

Тематичний і календарний план вивчення дисципліни

| № тижня | Тема лекції* | Тема практичного заняття* | Тема лабораторної роботи* | Самостійна робота студентів | | |
|---------|---|---------------------------|--|--|------|------------|
| | | | | Зміст | Год. | Література |
| 1. | Локальні та глобальні змінні. Підпрограми та їх аргументи. Визначення даних. Модификатор const. Модификатор volatile. Перевантаження функцій. [1, 3, 4]. | | Основні відмінності мови програмування C++ від Сі [1,4]. | Опрацювання лекційного матеріалу. Підготовка до практичної роботи №1 | 5 | [1-10] |
| 2. | Потоки. Робота з файлами [2, 3, 6]. | | | | 5 | [1-10] |
| 3. | Поняття класу в C++. Поняття об’єкту та об’єктно-орієнтоване програмування. Визначення методів класу поза класом. Методи класу. Модифікатори доступу. Використання оператора глобального дозволу для елементів класу. [1, 5, 7] | | Класи, методи, оператори new, delete в C++[1-4]. | Опрацювання лекційного матеріалу. Підготовка до захисту практичної роботи №1. Підготовка до практичної роботи №2 | 5 | [1-10] |
| 4. | Динамічний розподіл пам’яті. Ініціалізація виділеної пам’яті. Виділення пам’яті для масивів. Виділення пам’яті для об’єктів класів. [1, 8]. | | | | 5 | [1-10] |
| 5. | Конструктори. Конструктори за | | Конструктори та деструктори, | Опрацювання лекційного | 5 | [1-10] |

| | | | | | | |
|----|---|--|--|---|---|---------|
| | <p>замовчуванням. Конструктори з параметрами. Приватні конструктори. Конструктори копіювання. Деструктори. Загальнодоступні деструктори. Приватні деструктори. [1, 3, 4, 8]</p> | | <p>статичні функції та елементи даних, дружні функції та класи, конструктор копіювання в C++ [2,6,7,10].</p> | <p>матеріалу. Підготовка до захисту практичної роботи №2. Підготовка до практичної роботи №3</p> | | |
| 6. | <p>Статичні функції та елементи даних. Дружні функції. Визначення друзів класу. Властивості дружніх функцій. Приватні елементи і друзі. Використання статичних функцій-елементів. Ключове слово this [5, 6, 9]</p> | | | | 5 | [1-10] |
| 7. | <p>Поняття наслідування. Просте наслідування. Наслідування і захищені члени. Управління доступом до членів базового класу. Розв'язання конфлікту імен. Керування доступом до членів базового класу. Захищене наслідування. Конструктори, деструктори і наслідування [1, 6, 10, 11].</p> | | <p>Наслідування, множинне наслідування [2,6,7,10].</p> | <p>Опрацювання лекційного матеріалу. Підготовка до захисту практичної роботи №3. Підготовка до практичної роботи №4</p> | 5 | [11-12] |
| 8. | <p>Наслідування. Передача параметрів конструктору базового класу. Надання доступу при наслідуванні. Множинне наслідування. Віртуальні класи при наслідуванні. [2,6,7,10].</p> | | | | 5 | [11-12] |
| 9. | <p>Віртуальні функції та невіртуальні функції. Виклик віртуальної</p> | | <p>Поліморфізм, віртуальні функції, абстрактні класи</p> | <p>Опрацювання лекційного матеріалу. Підготовка до</p> | 5 | [1-12] |

| | | | | | | |
|-----|---|--|--|--|---|-------|
| | функції за допомогою посилання на об'єкт базового класу. Наслідування атрибуту virtual. поліморфізм. [2,6,7,11]. | | та виняткові ситуації. [1-4,10]. | захисту практичної роботи №4. Підготовка до практичної роботи №5 | | |
| 10. | Абстрактні класи. Застосування віртуальних функцій. Чисто віртуальні функції. Віртуальні деструктори. Віртуальні конструктори. [2,6,7,10]. | | | | 5 | [1-6] |
| 11. | Створення операторної функції-члена. Перевантаження скорочених операторів присвоєння. Обмеження на перевантажені оператори. Створення префіксної і постфіксної форм операторів інкрементації і декрементації. Перевантаження операторів за допомогою дружніх функцій. Застосування дружніх функцій для перевантаження операторів "++" і "--". Перевантаження операторів new і delete. Перевантаження деяких спеціальних операторів. [1,4,7,10]. | | Перевантаження операторів, шаблони в C++ [1,9,10]. | Опрацювання лекційного матеріалу. Підготовка до захисту практичної роботи №5. Підготовка до практичної роботи №6 | 5 | [1-6] |
| 12. | Шаблони функцій. Шаблони класів. Оголошення об'єктів, що базуються на шаблоні класу. [2,6,7,10]. | | | | 5 | [1-6] |
| 13. | Стандартна бібліотека шаблонів. | | Бібліотека STL. Контейнери, | Опрацювання лекційного | 5 | [1-6] |

| | | | | | | |
|-----|---|--|--|---|---|-------|
| | Складові STL. Класи-контейнери. [1,9,12]. | | літератори, предикати в C++. [1,2,6,7]. | матеріалу. Підготовка до практичної роботи №7. Підготовка до захисту практичної роботи №6. | | |
| 14. | Асоціативні контейнери. Алгоритми. Предикати. Застосування функторів. [1,9,12]. | | | | 5 | [1-6] |
| 15. | Створення подій. Обробка подій в C++. [6,7,13]. | | Windows Form Applications. [6,9,10] | Опрацювання лекційного матеріалу. Підготовка до практичної роботи №7. | 4 | [1-6] |
| 16. | Тема 9. Обробка подій. Компоненти, що утворюють інтерфейс між користувачем і додатком. Простір імен System. Основні компоненти. Властивості компонентів. [3, 9, 14]. | | | | 4 | [1-6] |
| 17. | Основи обробки виняткових ситуацій. Перехоплення класів виняткових ситуацій. Обробка похідних виняткових ситуацій. Обмеження виключних ситуацій. Функції terminate() і unexpected(). [1, 8, 15] | | | Опрацювання лекційного матеріалу. Підготовка до захисту практичної роботи №8. Індивідуальне оцінювання проекту колеги. Колективне оцінювання проекту одного з колег. | 4 | [1-6] |

Примітка: * Лекції, практичні заняття проводяться по дві години; послідовність проведення занять визначається розкладом (може не відповідати нумерованим тижням)

Політика дисципліни.

Організація освітнього процесу з дисципліни відповідає вимогам положень про організаційне і навчально-методичне забезпечення освітнього процесу, освітній програмі та навчальному плану. Студент зобов'язаний відвідувати лекції, лабораторні заняття згідно з розкладом, не запізнюватися на заняття, завдання виконувати відповідно до графіка. Пропущене лабораторне заняття студент зобов'язаний опрацювати самостійно у повному обсязі і відзвітувати перед викладачем не пізніше, ніж за тиждень до чергової атестації. До лабораторних занять студент має підготуватися за відповідною темою і проявляти активність. Набуті особою знання з дисципліни або її окремих розділів у неформальній освіті зараховуються відповідно до Положення про порядок перезарахування результатів навчання у ХНУ (<http://khnu.km.ua/root/files/01/06/03/006.pdf>).

Критерії оцінювання результатів навчання.

Кожний вид роботи з дисципліни оцінюється за **чотирибальною** шкалою. Семестрова підсумкова оцінка визначається як середньозважена з усіх видів навчальної роботи, виконаних і зданих позитивно з врахуванням коефіцієнта вагомості. Вагові коефіцієнти змінюються залежно від структури дисципліни і важливості окремих її видів робіт. При оцінюванні знань студентів використовуються різні засоби контролю, зокрема: усне опитування перед допуском до виконання лабораторної роботи – здійснюється на їх початку; засвоєння теоретичного матеріалу з тем перевіряється тестовим контролем; якість виконання, набуття теоретичних знань і практичних навичок перевіряється шляхом захисту кожної лабораторної роботи та індивідуального завдання згідно з робочою програмою дисципліни і робочим навчальним планом.

Структурування дисципліни за видами робіт і оцінювання результатів навчання студентів денної форми навчання у семестрі за ваговими коефіцієнтами

| | | | | | | | | | |
|-----------------------|---|---|---|-----|---|---|---|-------------------------------------|----------------------|
| Аудиторна робота | | | | | | | | Самостійна, індивідуальна робота | Підсумковий контроль |
| IX семестр | | | | | | | | | |
| Лабораторні роботи №: | | | | | | | | Тестовий контроль: | Іспит |
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | T 1-16 | Підсумкова КР |
| ВК: | | | | 0,4 | | | | 0,2 | 0,4 |

Примітка: ВК – ваговий коефіцієнт.

Співвідношення інституційної шкали оцінювання і шкали оцінювання ЄКТС

| Оцінка ЄКТС | Інституційна шкала балів | Інституційна оцінка | Критерії оцінювання |
|-------------|--------------------------|---------------------|--|
| A | 4,75-5,00 | 5 | Зараховано Відмінно – глибоке і повне опанування навчального матеріалу і виявлення відповідних умінь та навичок. Добре – повне знання навчального матеріалу з кількома незначними помилками. Добре – в загальному правильна відповідь з двома-трьома суттєвими помилками. Задовільно – неповне опанування програмного матеріалу, але достатнє для практичної діяльності за професією. Задовільно – неповне опанування програмного матеріалу, що задовольняє мінімальні критерії оцінювання |
| B | 4,25-4,74 | 4 | |
| C | 3,75-4,24 | 4 | |
| D | 3,25-3,74 | 3 | |
| E | 3,00-3,24 | 3 | |
| FX | 2,00-2,99 | 2 | Незараховано Незадовільно – безсистемність одержаних знань і неможливість продовжити навчання без додаткових знань з дисципліни Незадовільно – необхідна серйозна подальша робота і повторне вивчення дисципліни. |
| F | 0,00-1,99 | 2 | |

Питання для підсумкового контролю з дисципліни

- Загальні відомості про програми на C++. Структура проектів в C++.
- Елементи мови. Службові слова. Імена. Коментарі.
- Розміщення даних у пам'яті. Тип даних. Адресний вираз.
- Функції. Область існування імені. Область видимості та простору імен.
- Типи. Базові стандартні типи. Константи. Змінні.
- Операції C++. Арифметичні операції. Операції присвоєння. Операції відношення та еквівалентності. Логічні операції.
- Операції адресації та непрямой адресації.
- Пріоритет і асоціативність операцій. Перегрузка операцій.
- Оператори передачі управління. Умовний оператор множинного вибору.
- Оператори циклів: for, do...while, while.
- Переривання циклу: оператори break, continue, return, функція Abort.
- Умовна компіляція. Операції препроцесора.
- Масиви у C++. Операції з масивами.
- Правила роботи з масивами.
- Структури та об'єднання. Структури в стилі C++.

16. Об'єднання. Об'єднання, що не мають імені. Доступ до членів-даних структури, об'єднання.
17. Екземпляри структури, об'єднання. Масиви структур.
18. Вказівники, зсилки та функції в C++.
19. Локальний та динамічний розподіл пам'яті. Функції malloc, calloc, realloc, free.
20. Оператори new і delete. Динамічне розміщення об'єктів та вказівників.
21. Правила роботи з вказівниками. Розіменування вказівників.
22. Оператори new[], оператори delete[].
23. Функції в C++. Передача аргументів та повернення результату.
24. Застосування при передачі параметрів специфікації const. Параметри зі значеннями по умовчанняю.
25. Перегрузка функцій. Приведення типів.
26. Функції-члени класу. Передача у функції змінного числа параметрів.
27. Вбудовані функції inline.
28. Шаблони функцій.
29. Область видимості функцій. Правила, що визначають область видимості.
30. Функції main(). Аргументи функції main. Аргументи командного рядка.
31. Основні поняття ООП. Інкапсуляція, поліморфізм, наслідування і засоби їх реалізації.
32. Об'єкт, властивості, методи, події.
33. Класи. Визначення класу. Ідентифікатори класу. Тіло класу.
34. Приховування інформації. Ієрархії класів та наслідування.
35. Інкапсуляція. Керування доступом до класу. Приватні, загальнодоступні, захищені члени класу.
36. Класи пам'яті для об'єктів класів. Область видимості класу. Порожні класи.
37. Вкладені класи. Правила доступу для вкладених класів.
38. Екземпляри класу. Використання членів даних. Статичні члени-данні.
39. Об'єкти класу як члени даних. Вказівники як члени даних.
40. Використання функцій-членів. Прості, статичні функції члени.
41. Конструктори. Деструктори.
42. Дружні функції. Властивості дружніх функцій.
43. Шаблони класів. Вкладені шаблонні класи.
44. Наслідування. Обмеження наслідування. в C++. Просте наслідування.
45. Специфікатори доступу базових класів. Класи для наслідування. Аргументи передані в базовий клас.
46. Порядок виклику конструкторів. Порядок виклику деструкторів.
47. Перетворення типів у похідних класах. Дозвіл області видимості.
48. Множинне наслідування. Оголошення класу з декількома базовими класами. Виклик конструкторів базових класів.
49. Використання віртуальних базових класів. Використання віртуальних і невіртуальних базових класів разом.
50. Використання перетворення типів. Правила виклику функцій базових класів.
51. Використання дозволу видимості при множинному наслідуванні.
52. Поліморфізм. Раннє та пізнє зв'язування. Віртуальні функції. Перевизначення функцій.
53. Абстрактні класи. Обмеження віртуальних функцій. Віртуальні оператори.
54. Поліморфізм при простому наслідуванні. Поліморфізм при множинному наслідуванні.
55. Виклик поліморфних функцій базового класу. Віртуальні функції та ієрархії класифікації. Виклик віртуальних функцій у конструкторах.
56. Потік C++. Потоки як узагальнені фільтри. Стандартний потоковий ввід-вивід класів користувача.
57. Потоки і файловий ввід-вивід. Використання текстових файлів для введення. Вивід текстових файлів.
58. Обробка подій. Об'єктно-орієнтована програма, яка керована подіями. Методи опрацювання подій.
59. Шаблони функцій і класів. Основні властивості шаблонів класів. Компонентні функції параметризованих класів.
60. Стандартна бібліотека шаблонів. Склад STL. Класи-контейнери. Асоціативні контейнери. Алгоритми.

9. МЕТОДИЧНЕ ЗАБЕЗПЕЧЕННЯ

Навчальний процес з дисципліни забезпечений необхідними навчально-методичними розробками в модульному середовищі.

10. РЕКОМЕНДОВАНА ЛІТЕРАТУРА

1. Шилдт. Герберт. C++. руководство для начинающих, 6-е п л анис.: Пер. с англ. — М.: Издательский дом "Вильямс", 2015. — 672 с.: ил. — Парад. тит. англ. ISBN 5-8459-0840-X (рус.)
2. Campbell Parallel Programming with Microsoft® Visual C++® / Campbell. - Москва: Гостехиздат, 2015. - 784 с.
3. Балена, Франческо Современная практика программирования на Microsoft Visual Basic и Visual C# / Франческо Балена, Джузеппе Димауро. - М.: Русская Редакция, 2015. - 640 с.
4. Боровский, А. C++ и Pascal в Kylix 3. Разработка интернет-приложений и СУБД / А. Боровский. - М.: БХВ-Петербург, 2015. - 544 с.
5. Давыдов, В. Visual C++. Разработка Windows-приложений с помощью MFC и API-функций / В. Давыдов. - М.: БХВ-Петербург, 2014. - 576 с.
6. Довбуш Г. Visual C++ на примерах / Галина Довбуш, Анатолий Хомоненко. - М.: БХВ-Петербург, 2017. - 528 с.

7. Зиборов, В. MS Visual C++ 2015 в среде .NET / В. Зиборов. - М.: Питер, 2015. - 320 с.
8. Кетков, Юлий Практика программирования: Visual Basic, C++ Builder, Delphi. Самоучитель (+ дискета) / Юлий Кетков, Александр Кетков. - М.: БХВ-Петербург, 2012. - 464 с.
9. Мешков, А. Visual C++ и MFC / А. Мешков, Ю. Тихомиров. - М.: БХВ-Петербург, 2013. - 546 с.
10. Неформальное введение в C++ и Turbo Vision. - Москва: ИЛ, 2017. - 384 с.
11. Панюкова, Т. А. Языки и методы программирования. Создание простых GUI-приложений с помощью Visual C++. Учебное пособие / Т.А. Панюкова, А.В. Панюков. - Москва: Мир, 2015. - 144 с.
12. Пахомов Б. C/C++ и MS Visual C++ 2012 для начинающих / Борис Пахомов. - Москва: СИНТЕГ, 2015. - 518 с.
13. Полубенцева, М. C/C++. Процедурное программирование / М. Полубенцева. - М.: БХВ-Петербург, 2014. - 448 с.
14. Поляков, А. Методы и алгоритмы компьютерной графики в примерах на Visual C++ / А. Поляков, В. Брусенцев. - М.: БХВ-Петербург, 2015. - 560 с.
15. Роберт, С. Сикорд Безопасное программирование на C и C++ / Роберт С. Сикорд. - Москва: РГГУ, 2016. - 496 с.
16. Секунов, Н. Программирование на C++ в Linux / Н. Секунов. - М.: БХВ-Петербург, 2016. - 425 с.
17. Сидорина Т. Самоучитель Microsoft Visual Studio C++ и MFC / Т. Сидорина. - М.: "БХВ-Петербург", 2018. - 848 с.

Разработчик:


к.т.н., доц. Лисенко С.М.

Погоджено:

Зав. каф. КИСП:


д.т.н., проф. Говорущенко Т.О.

Гарант ОПП «АКИТ»:


к.т.н., доц. Форкун Ю.В.