

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ ТОРГОВЕЛЬНО-ЕКОНОМІЧНИЙ
УНІВЕРСИТЕТ**

СИСТЕМА УПРАВЛІННЯ ЯКІСТЮ

**Система забезпечення якості освітньої діяльності та якості вищої
освіти**

сертифікована на відповідність ДСТУ ISO 9001:2015 / ISO 9001:2015

Кафедра цифрової економіки та системного аналізу

Кафедра комп'ютерних наук та інформаційних систем

ЗАТВЕРДЖЕНО

вченою радою

(пост. в. *І* від «*25*» *2020*р.)

Ректор

А. А. Мазаракі



**ІНСТРУМЕНТАЛЬНІ ЗАСОБИ
ПРИКЛАДНОГО ПРОГРАМУВАННЯ/
APPLICATION PROGRAMMING TOOLS**

**ПРОГРАМА /
COURSE SUMMARY**

КИЇВ 2020

Розповсюдження і тиражування без офіційного дозволу КНТЕУ заборонено

Автори: В.В.Кулаженко канд.екон.наук доц.
Т.В. Томашевська кан.тех. наук. доц.
С.А. Міценко кан.тех. наук. доц.

Програму розглянуто і затверджено на засіданні кафедри комп'ютерних наук та інформаційних систем «18» червня 2020р., протокол № 18

Рецензенти: А.А. Роскладка, док. екон. наук, проф.
О.І.Пурський док.фіз-мат наук проф.
С.П. Кудрявцева, Провідний науковий співробітник
Міжнародного науково-навчального центру інформаційних технологій та систем НАН та МОН України,
канд. техн. наук.

**ІНСТРУМЕНТАЛЬНІ ЗАСОБИ
ПРИКЛАДНОГО ПРОГРАМУВАННЯ/
APPLICATION PROGRAMMING TOOLS**

**ПРОГРАМА /
COURSE SUMMARY**

ВСТУП

Програма дисципліни «Інструментальні засоби прикладного програмування» призначена для здобувачів освітнього ступеня «Молодший бакалавр», галузі знань 12 «Інформаційні технології», спеціальності 122 «Комп'ютерні науки», спеціалізації «Комп'ютерні науки».

Розроблена програма складається з таких розділів:

1. Мета, завдання та предмет дисципліни.
2. Передумови вивчення дисципліни як компоненти освітньої програми.
3. Результати вивчення дисципліни.
4. Зміст дисципліни.
5. Список рекомендованих джерел.

1. МЕТА, ЗАВДАННЯ ТА ПРЕДМЕТ ДИСЦИПЛІНИ

Метою вивчення дисципліни “Інструментальні засоби прикладного програмування” є формування у студентів навичок оволодіння технологіями обробки простих та структурованих даних, опанування технологій структурного, модульного та об'єктно-орієнтованого програмування на базі мови програмування C++.

Завданням вивчення дисципліни є теоретична та практична підготовка студентів з таких питань:

- парадигми програмування (структурного, модульного, об'єктно-орієнтованого) та засоби сучасних мов програмування для реалізації різних концепцій;
- засоби мов програмування для реалізація розгалужених та циклічних алгоритмів;
- використання функцій та окремих модулів користувача;
- реалізація класичних інформаційних структур (списків, дерев) з використанням статичного та динамічного розподілу пам'яті;
- базові технології проектування процесів пошуку та побудови впорядкованих даних у лінійних списках та деревовидних структурах;
- використання класів для реалізації парадигми об'єктно-орієнтованого програмування;
- засоби створення ієрархічної об'єктної структури з використанням базового об'єкта, подальше розширення його властивостей з використанням статичних та віртуальних методів;
- сучасні інструментальні засоби для створення прикладного програмного забезпечення. Автоматизація програмування: технологія RAD (Rapid Application Development);

- особливості програмування в операційних системах родини Windows. Система управління повідомленнями. Структура програми для Windows;

- програмування графічного інтерфейсу користувача, ресурси прикладної програми. Функції API (Application Programming Interface);

- застосування бібліотеки класів MFC (Microsoft Foundation Classes) як інструменту для створення прикладних програм для Windows;

- специфікація COM (Component Object Model) як основа технології створення компонент ActiveX. Поняття інтерфейсу компоненти, поліморфізм, таблиця віртуальних функцій, множинні екземпляри;

- динамічне компонування прикладної програми.

Предметом дисципліни є знання у сфері технологій програмування простих та більш складних алгоритмів реалізованих засобами мови програмування C++.

2. ПЕРЕДУМОВИ ВИВЧЕННЯ ДИСЦИПЛІНИ ЯК ВИБІРКОВОЇ КОМПОНЕНТИ ОСВІТНЬОЇ ПРОГРАМИ

Знання:

предмету і сутності комп'ютерних технологій обробки даних;

особливостей роботи операційної системи Windows;

основ вищої математики та дискретного аналізу;

основ методології математичного моделювання економічних процесів;

механізмів застосування теоретичних методів і моделей у відображенні економічних процесів;

Вміння:

- вільно працювати з файловою структурою операційної системи Windows;

- працювати з основними програмами пакету MS Office;

- розуміти математичні моделі структурованих даних;

3. РЕЗУЛЬТАТИ ВИВЧЕННЯ ДИСЦИПЛІНИ

Дисципліна "Інструментальні засоби прикладного програмування", як обов'язкова компонента освітньої-професійної програми, забезпечує оволодіння студентами загальними та фаховими компетентностями і досягнення ними програмних результатів навчання за відповідною освітньо-професійною програмою:

Комп'ютерні науки (ОС молодший бакалавр)

Номер в освітній програмі	Зміст компетентності	Номер теми, що розкриває зміст компетентності
<i>Загальні компетентності за освітньо-професійною програмою</i>		
ЗК 2	Здатність застосовувати знання у практичних ситуаціях.	1.1.,2.1.
ЗК 3	Знання та розуміння предметної області та розуміння професійної діяльності.	1.1,1.2,1.3
ЗК 4	Здатність спілкуватися державною мовою як усно, так і письмово в термінології предметної області.	1.3, 2.3
ЗК 7	Здатність до пошуку, оброблення та аналізу інформації з різних джерел.	2.7.
<i>Фахові компетентності за освітньо-професійною програмою</i>		
СК 6	Здатність застосовувати методології, технології та інструментальні засоби для розробки програмних систем, продуктів і сервісів інформаційних технологій.	1.2, 2.5
СК 8	Здатність проектувати та розробляти програмне забезпечення із застосуванням різних парадигм програмування: узагальненого, об'єктно-орієнтованого, функціонального, логічного, з відповідними моделями, методами й алгоритмами обчислень, структурами даних і механізмами управління.	2.3.,2.4.,2.5.

<i>Програмні результати навчання за освітньо-професійною програмою</i>		
ПР 4	Використовувати високорівневі мови програмування, обчислювальні методи і алгоритми для розробки програмних засобів обробки даних.	1.1, 1.2, 1.3, 2.1, 2.2, 2.3
ПР 8	Використовувати інструментальні засоби проектування концептуальних, логічних та фізичних моделей баз даних, створювати бази даних, розробляти та оптимізувати запити до них, у тому числі із застосуванням мов програмування.	2.6, 2.7.,2.8.

4. ЗМІСТ ДИСЦИПЛІНИ

РОЗДІЛ 1. Основи програмування за допомогою Application Programming Interface (API).

Тема 1.1. Знайомство з середовищами розроблення програм

Види програмного забезпечення для розроблення програм в операційній системі Windows. Знайомство з інструментальними засобами MS Visual Studio. Майстри для створення проектів різних типів. Поняття проекту в MS Visual C++. Основні структурні компоненти проекту. Аналіз програмного забезпечення, що створюється майстрами MS Visual C++. Компіляція та запуск програм на виконання. Інтерактивні засоби Microsoft Visual C++ для налагодження програм.

Список рекомендованих джерел

Основний: 1,2,3.

Додатковий: 6, 7.

Інтернет-ресурси: 10,12,15.

Тема 1.2. Основи програмування в операційній системі Windows

Структура програми Application Programming Interface (API) для Windows. Функція WinMain. Вікно як основний об'єкт програми для Windows. Клас вікна та його реєстрація. Типи вікон. Створення вікна

програми. Цикл оброблення повідомлень. Механізм передачі повідомлень. Структура віконної процедури.

Список рекомендованих джерел

Основний: 3,5.

Додатковий: 6,8

Інтернет-ресурси: 11,12, 15

Тема 1.3. Керування виведенням інформації в системі Windows

Контекст пристрою. Види графічних об'єктів. Режими відображення та різні системи координат. Робота з текстом. Функції API для керування графічним виводом.

Список рекомендованих джерел

Основний: 1,2,5.

Додатковий: 7,8.

Інтернет-ресурси: 10, 12, 13

РОЗДІЛ 2. ЗАСОБИ ДЛЯ РОЗРОБЛЕННЯ ПРОГРАМ У СИСТЕМАХ WINDOWS З ЗАСТОСУВАННЯМ БІБЛІОТЕКИ КЛАСІВ MFC.

Тема 2.1. Створення MFC-програм

Призначення бібліотеки класів MFC (Microsoft Foundation Classes). Аналіз класів для програм з одним та декількома документами. Поліморфізм та віртуальні функції, перевантаження віртуальних функцій в успадкованих класах. Призначення та структура карти повідомлень (message maps). Макроси, що використовуються в карті повідомлень. Створення та редагування карти повідомлень за допомогою майстра. Створення функцій для оброблення повідомлень Windows (Windows message handler). Поняття команди (command). Створення функцій для оброблення команди.

Список рекомендованих джерел

Основний: 1, 3

Додатковий: 8,9

Інтернет-ресурси: 12, 15

Тема 2.2. Створення графічного інтерфейсу користувача в MFC-програмах

Формування ресурсів вікна діалогу. Створення класу вікна діалогу. Обмін даними між програмою та вікном діалогу за допомогою змінних класу. Робота з елементами управління різних типів. Формування ресурсів меню. Створення функції для оброблення команди меню. Формування ресурсів панелей інструментів. Створення функції для оброблення команди кнопки панелі інструментів.

Список рекомендованих джерел

Основний: 3, 4

Додатковий: 8

Інтернет-ресурси: 14, 15

Тема 2.3. Виведення даних у вікно документа в MFC-програмах

Призначення класу CPaintDC. Клас відображення CView. Використання функцій класу для виведення інформації документа. Робота з графічними об'єктами в MFC-програмах. Графічні примітиви.

Призначення полів структури Windows LOGFONT. Використання олівців (pens) та пензликів (brushes).

Список рекомендованих джерел

Основний: 3, 5

Додатковий: 6,8

Інтернет-ресурси: 10, 13

Тема 2.4. Створення додатку на основі діалогових вікон

Призначення класу додатку CApp, структура файлу оголошення класу та файлу функцій. Призначення та реалізація класів відображення CDialog. Перевантаження функцій та створення повідомлень і функцій обробки повідомлень. Створення власних класів на основі класів MFC. Застосування класів CMetaFileDC збереження документу при роботі додатку.

Список рекомендованих джерел

Основний: 3,5

Додатковий: 6,8

Інтернет-ресурси: 10, 13

Тема 2.5. Створення додатку на основі концепції DocView

Призначення класу додатку CApp, класів CMainFrame, CDoc та CView структура файлу оголошення класів та файлу функцій. Реалізація класів відображення CDialog та CView. Перевантаження функцій та

створення повідомлень і функцій обробки повідомлень. Створення власних класів на основі класів MFC. Застосування класів похідних вікон CChildFrame та імплементації контейнеру CCtrlItem. Застосування класу CDoc для збереження документу додатку.

Список рекомендованих джерел

Основний: 3, 5

Додатковий: 7,8

Інтернет-ресурси: 13, 14, 15

Тема 2.6. Робота з базами даних у MFC-програмах

Сучасні технології доступу до реляційних баз даних. Технологія ODBC. Використання класу CDataBase для оброблення бази даних. Створення вікон діалогу для відображення записів таблиць бази даних. Використання класів CRecordset, CRecordView для роботи з записами таблиць бази даних. Використання SQL (Structured Query Language) для одержання інформації із декількох таблиць бази даних.

Список рекомендованих джерел

Основний: 3

Додатковий: 7,8

Інтернет-ресурси: 11,13

Тема 2.7. Основи моделі компонентних об'єктів Microsoft

Основні поняття технології ActiveX- Приклади. Модель компонентних об'єктів (COM - Component Object Model) як основа технології ActiveX. Поняття компоненти. Архітектура прикладної програми на основі компонент. Поняття інтерфейсу компоненти. Взаємодія з компонентою за допомогою її інтерфейсів. Призначення інтерфейсу IUnknown. Віртуальні функції інтерфейсу IUnknown. Запит інтерфейсу компоненти за допомогою функції QueryInterface. Керування життям компоненти. Функції AddRef та Release.

Список рекомендованих джерел

Основний: 3, 5

Додатковий: 6,8

Інтернет-ресурси: 11, 13

Тема 2.8. Динамічне компонування програми

Поняття динамічної бібліотеки DLL (Dynamic Link Library). Переваги динамічного компонування програми. Експорт функції із DLL. Ро-

збиття монолітної програми на файли клієнта та компоненти. Розміщення компоненти в динамічній бібліотеці.

Список рекомендованих джерел

Основний: 1, 3, 5

Додатковий: 6,7

Інтернет-ресурси: 11, 15

5. СПИСОК РЕКОМЕНДОВАНИХ ДЖЕРЕЛ

Основний¹

1. Грицюк Ю Програмування мовою C++ / Ю.Грицюк, Т.Рак навчальний посібник. – Львів : Вид-во Львівського ДУ БЖД, 2011. – 292 с.

2. Белов Ю. Вступ до програмування мовою C++. Організація обчислень: навч. посіб. / Ю.Белов, Т.Карнаух, Ю.Коваль, А.Ставровський. – К. : Видавничо-поліграфічний центр "Київський університет", 2012. – 175 с. с.: іл. (укр.)

3. Stephen D.Gilbert, Bill McCarty Visual C++ Programming Blue Book. — The Coriolis Group., Inc.,2015. — 14455 N Hayden Drive., Suite 200 Arizona 85260 USA.

4. Sedgewick R., Algorithms in C++ :Fundamentals, data structur, sorting, serching / Princeton University 2011.– 1056 с.

5. A Tour of C++ (2nd Edition) (C++ In-Depth Series) / by Bjarne Stroustrup. – Addison-Wesley Professional, July 9, 2018. – 256 p.

Додатковий

6. Programming: Principles and Practice Using C++ (2nd Edition) / by Bjarne Stroustrup. – Addison-Wesley Professional, May 25, 2014. – 1312 p.

7. Practical Programming for Strength Training / by Mark Rippetoe, Andy Bake. – The Aasgaard Company, January 14, 2014. – 256 p.

8. Effective Modern C++: 42 Specific Ways to Improve Your Use of C++11 and C++14 1st Edition / by Scott Meyers. - O'Reilly Media, Incorporated, December 5, 2014. – 334 p.

Інтернет ресурси

¹ Курсивом позначені видання, що присутні у бібліотеці КНТЕУ

9. Український сайт, присвячений використанню та українізації операційної системи Linux. [Електронний ресурс]. - Режим доступу : <http://linux.org.ua>
10. Книжки з методик програмування на C++. [Електронний ресурс]. - Режим доступу : <http://progbook.net/ss>
11. Довідник мови програмування C++ [Електронний ресурс]. – Режим доступу: <https://msdn.microsoft.com/uk-ua/library/3bstk3k5.aspx>.
12. Сайт про мову програмування C++ [Електронний ресурс]. – Режим доступу: <http://www.cplusplus.com/>.
13. Довідник по IDE VisualStudio [Електронний ресурс]. – Режим доступу: <https://msdn.microsoft.com/>.
14. Сайт підтримки бібліотеки NumPy [Електронний ресурс]. – Режим доступу : <http://www.numpy.org>.
15. Книжки по методикам програмування на C++. [Електронний ресурс]. – Режим доступу: <http://progbook.net/ss/>