

**Відокремлений структурний підрозділ  
«Конотопський індустріально-педагогічний фаховий коледж СумДУ»  
Циклова комісія комп'ютерних та математичних дисциплін**



**ЗАТВЕРДЖУЮ**  
Заступник директора  
з навчальної роботи

Кравченко О.О.  
2023 року

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

**Об'єктно-орієнтоване програмування**

(назва навчальної дисципліни)

Галузь знань 01 Освіта/Педагогіка

(шифр і назва напрямку підготовки)

спеціальність 015 Професійна освіта

(шифр і назва спеціальності)

Спеціалізація 39 Цифрові технології

відділення Професійної освіти

(назва відділення)

Робоча програма «Об'єктно-орієнтоване програмування » для студентів за галуззю знань за галуззю знань 01 Освіта/Педагогіка спеціальність 015 Професійна освіта  
«30» серпня 2023 року - 19 с.

Розробники: викладач ВСП «Конотопський індустріально-педагогічного фахового коледжу СумДУ Бараболіков В.М..

Робоча програма затверджена на засіданні циклової комісії Комп'ютерних та математичних дисциплін  
Протокол від. «30» серпня 2023 року № 1

Голова циклової комісії  (Бараболіков В.М.)  
(підпис)

«\_\_» серпня 2023 року

Схвалено методичною радою ВСП «Конотопського індустріально-педагогічного фахового коледжу СумДУ.

Протокол від «\_\_» \_\_\_\_\_ 20\_\_ року № \_\_\_\_

«\_\_» \_\_\_\_\_ 20\_\_ року Голова \_\_\_\_\_ (Кравченко О. О.)  
(підпис)

## 1. Опис навчальної дисципліни

Найменування показників	Галузь знань, напрям підготовки, освітньо-кваліфікаційний рівень	Характеристика освітнього компоненту		
		денна форма навчання		
Кількість кредитів-3	Галузь знань 01 «Освіта/Педагогіка» (шифр і назва)	Нормативна		
	Напрямок підготовки  (шифр і назва)			
Розділів – 3	Спеціальність <u>015 Професійна освіта</u> Спеціалізація <u>Цифрові технології</u>	<b>Рік підготовки:</b>		
Тем – 7		1-й		
Індивідуальне науково-дослідне завдання		<b>Семестр</b>		
Загальна кількість годин - 90		3-й	4-й	5-й
Годин для денної форми навчання: аудиторних – 64 самостійної роботи студента - 26		<b>Лекції</b>		
		36		
	<b>Практичні</b>			
		14		
	<b>Лабораторні</b>			
		14		
	<b>Самостійна робота</b>			
		26		
<b>Індивідуальні завдання</b>				
Вид контролю: залік				

### Примітка.

Співвідношення кількості годин аудиторних занять до самостійної і індивідуальної роботи становить:

для денної форми навчання – **1 : 0,9**

для заочної форми навчання –

## 2. Мета та завдання навчальної дисципліни

Об'єктно-орієнтоване програмування (ООП) являється новим розумінням того, що власне називається обчислювальним, а також те, як ми можемо структурувати інформацію в комп'ютері. Цей еволюційний крок, природнім шляхом впливає з попередньої історії.

**Предметом вивчення** курсу “Об'єктно-орієнтоване програмування” є основні принципи об'єктно-орієнтованого підходу до програмування, технології візуального проектування та подійного програмування в інтегрованому середовищі розробки додатків С ++ .

**Мета курсу** – розуміння сучасних об'єктно-орієнтованих технологій програмування, опанування інтегрованого середовища розробки додатків С ++ , поштовх до подальшого вивчення цієї та інших мов програмування.

Переваги ООП в повній мірі проявляються лише при розробці достатньо складних програм. Більш того, інкапсуляція надає об'єктам зовсім особливу властивість «самостійність», максимальну незалежність від інших частин програми. Правильно сконструйований об'єкт володіє всіма необхідними даними і процедурами їх обробки, для успішної реалізації необхідних від нього дій. При розробці складних діалогових програм програміст повинен структурувати програму, так як тільки в цьому випадку він може розраховувати на успіх: «критичною вагою» неструктурованих програм являється об'єм в 1000—1200 рядків висхідного тексту — відладка неструктурованих програм більшого розміру звичайно проходять з великими труднощами. Структуровані програми ведуть, фактично, до розробки власної бібліотеки програмування — ось у цю мить до Вас на допомогу і приходять нові засоби ООП.

### **Вимоги щодо знань, вмінь, набутих внаслідок вивчення дисципліни.**

#### **Програмні результати навчання**

- принципи складання алгоритмів та складання блок-схем до алгоритмів;
- оператори вводу-виводу інформації правила їх використання;
- умовні оператори, оператори вибору варіанту та їх можливості при створенні програмних продуктів;
- можливості при створенні циклів на мові С ++; їх особливості їх використання;
- правила написання функцій процедур та класів у С ++;
- організацію роботи з файлами, організацію доступу до зовнішніх носіїв;
- використання динамічної пам'яті;
- графічні оператори, правила побудови растрових та розширених зображень;
- інтерфейс програмування С ++ , основні компоненти сторінок компонент, їх основні властивості та правила їх використання;
- роботу з базами даних в С ++ ;
- методику написання довідки та створення інсталяційних пакетів.



## Фахові компетентності

- Здатність застосовувати різні описи алгоритмів;
- Здатність визначати тип величини, описувати її мовою “С ++”;
- Здатність використовувати прості й складні умови під час побудови програм;
- Здатність описувати алгоритми розв’язання різних типів задач мовою програмування;
- Здатність складати й реалізовувати найпростіші лінійні, розгалуджені, циклічні алгоритми при опрацюванні табличних і рядкових величин, графічної інформації;
- Здатність використовувати середовище програмування С ++ ;
- Здатність вміло застосовувати компоненти різних палітр

## Місце освітнього компоненту в структурно-логічній схемі ОПШ та міжпредметні зв’язки:

Із курсом “Об’єктно-орієнтоване програмування” студенти ознайомлюються після вивчення наступних професійно-орієнтованих дисциплін: інформатика; дискретна математика, архітектура комп’ютера. Нижче подано перелік дисциплін, що можуть читатися після курсу “Об’єктно-орієнтоване програмування”, певним чином залежні від нього або перетинаються із ним: Основи програмування та алгоритмічні мови, Операційні системи; Архітектура та проектування програмного забезпечення; Аналіз вимог до програмного забезпечення; Якість програмного забезпечення; Моделювання програмного забезпечення; Безпека програм та даних.

### 3. Програма навчальної дисципліни

#### Розділ 1. Основні поняття об'єктно-орієнтованого програмування

##### Тема 1.1. Об'єктно-орієнтоване програмування: підхід до організації програми

Процедурні мови. Недоліки структурного програмування. Об'єктно-орієнтований підхід. Основні концепції об'єктно-орієнтованого програмування: класи, наслідування, поліморфізм.

*Література:* [1] с. 20 – 125

[2] с.151-264

Лабораторна робота № 1

Класи і об'єкти в с++

##### **Тема 1.2. Об'єкти та класи**

Класи та об'єкти. Визначення класу: дані класу, методи класу. Інкапсуляція – механізм приховування даних. Рівні доступу до членів класу. Клас як тип даних. Виклик методів класу. Конструктори. Автоматична ініціалізація. Конструктор за замовчуванням. Конструктор з параметрами. Деструктори.

Конструктор копіювання за замовчуванням. Перевантажені конструктори. Визначення методів класу поза класом. Статичні дані класу. Застосування статичних полів класу. Константні методи. Константні аргументи методів. Константні об'єкти.

*Література:* [1] с. 120 – 225

[2] с.151-164

Лабораторна робота № 2

Спадкоємство і віртуальні функції

Практична робота № 1

ієрархія об'єктів і група. Ітератори

Практична робота № 2

Обробка подій

##### **Тема 1.3. Наслідування**

Наслідування. Просте наслідування. Базовий та похідний класи. Визначення похідного класу. Доступ до базового класу. Підстановка конструкторів базового класу. Підстановка методів базового класу. Незмінність базового класу.

Ієрархія класів. Рівні наслідування. Комбінації доступу. Множинне наслідування. Конструктори при множинному наслідуванні. Невизначеність при множинному наслідуванні.

Поліморфізм і перевантаження.

*Література:* [1] с. 220 – 175

[3] с.151-264

Лабораторна робота № 3  
Перевантаження операцій

Практична робота № 3  
Шаблони функцій і класів

## **Розділ 2. Розробка об'єктно-орієнтованого програмного забезпечення**

### **Тема 2.1. Бібліотеки класів C ++ і модель візуальних компонентів**

Бібліотеки класів C ++ як приклад ООП. Бібліотека візуальних компонентів VCL.

Структура VCL. Службові класи. Класи компонентів. Абстрактні базові класи.

Класи GDI. Модель властивостей, методів та подій.

Класи стандартних компонентів. Класи системних компонентів.

*Література:* [1] с. 120 – 225

Лабораторна робота №4  
Потокові класи

Практична робота № 4  
СТАНДАРТНА БІБЛІОТЕКА ШАБЛОНІВ

### **Тема 2.2. Огляд класів компонентів VCL**

Візуальні та невізуальні компоненти. Клас форм і додатків. Основні властивості, методи та події. Класи системних діалогів. Метод Execute().

Основні елементи управління Windows. Компоненти для роботи з текстовою інформацією. Компоненти вибору зі списків. Компоненти – меню. Панелі та компоненти зовнішнього оформлення. 291

*Література:* [1] с. 201 – 256

Лабораторна робота №5  
Розробка додатків. Послідовність розробки та виконання додатків

Практична робота №5  
Програмування клавіатури. Реакція на події клавіатури

## **Розділ 3. Програмування графіки та баз даних**

### **Тема 3.1. Методи графічного програмування**

Поняття полотна та графічного курсору. Малювання за допомогою Canvas.

Основні методи креслення графічних примітивів. Компоненти Pen, Brush: основні властивості. Точка. Текст.



Графічний редактор Image Editor. Створення файла зображення. Створення піктограми для шаблону компонента в бібліотеці. Створення піктограми для кнопки. Робота з файлами ресурсів.

Робота з ілюстраціями. Компонент Image: основні властивості. Мультиплікація. Метод базової точки.

Мультимедіа. Компонент Animate: основні властивості. Компонент MediaPlayer: основні властивості. Відтворення звуку. Перегляд відеороликів. Створення анімації.

*Література:* [4] с. 20 – 225

#### Лабораторна робота №6

Робота зі списками. Організація багатосторінкових списків. Обробка текстових файлів

#### Лабораторна робота №6

Конструювання меню. Стандартні меню. Контекстні меню. Робота з дизайнером меню Menu Designer

### **Тема 3.2. Архітектура доступу до баз даних**

Поняття бази даних. Структура бази даних. Локальні та віддалені бази даних.

Механізм доступу до баз даних. Процесор баз даних (BDE). Псевдоніми BDE.

Архітектура доступу до даних, компоненти доступу. Основні властивості, методи, події. Розробка баз даних. Поля даних. Відображення даних. Редагування даних.

Вибір інформації з бази даних: пошук даних, запити, звіти.

*Література:* [5] с. 420 – 5 23

#### Лабораторна робота №7

Графіка у . Клас TCanvas

#### Практична робота №7

Розробка системи управління базою даних. Розробка бази даних



## 4. Структура навчальної дисципліни

Назви розділів і тем	Кількість годин											
	денна форма						Заочна форма					
	усього	у тому числі					усього	у тому числі				
		л	п	лаб	інд	с.р		л	п	лаб	інд	с.р
1	2	3	4	5	6	7	8	9	10	11	12	13
<i>Модуль № 1</i>												
<b>Розділ 1. Основи об'єктно-орієнтованого програмування</b>												
Тема 1.1. ООП: підхід до організації програми	6	2		2		2						
Тема 1.2. Об'єкти та класи	14	6	4	2		2						
Тема 1.3. Наслідування	12	6	2	2		2						
<b>Всього по розділу-1</b>	<b>32</b>	<b>14</b>	<b>6</b>	<b>6</b>		<b>6</b>						
<b>Розділ 2. Розробка об'єктно-орієнтованого програмного забезпечення</b>												
Тема 2.1. Бібліотека класів C++ і модель візуальних компонентів	15	6	2	2		5						
Тема 2.2. Огляд класів компонентів VCL	15	6	2	2		5						
<b>Всього по розділу-2</b>	<b>30</b>	<b>12</b>	<b>4</b>	<b>4</b>		<b>10</b>						
<b>Розділ 3. Програмування графіки та баз даних</b>												
Тема 3.1. Методи графічного програмування	15	6	2	2		5						
Тема 3.2. Архітектура доступу до баз даних	13	4	2	2		5						
<b>Всього по розділу-3</b>	<b>28</b>	<b>10</b>	<b>4</b>	<b>4</b>		<b>10</b>						

Назви розділів і тем	Кількість годин												
	денна форма						Заочна форма						
	усього	у тому числі					усього	у тому числі					
		л	п	лаб	інд	с.р		л	п	лаб	інд	с.р	
1	2	3	4	5	6	7	8	9	10	11	12	13	
<b>Всього</b>	<b>90</b>	<b>36</b>	<b>14</b>	<b>14</b>		<b>26</b>							

### 5. Теми семінарських занять

Не передбачені навчальною програмою

### 6. Теми практичних занять

№ зан.	Назва теми	Кількість годин
1.	<b>Практичне заняття № 1</b> Ієрархія об'єктів і група. Ітератори	2
2.	<b>Практичне заняття № 2</b> Обробка подій	2
3.	<b>Практичне заняття № 3</b> Шаблони функцій і класів	2
4.	<b>Практичне заняття № 4</b> Стандартна бібліотека шаблонів	2
5.	<b>Практичне заняття №5</b> Програмування клавіатури. Реакція на події клавіатури	2
6.	<b>Практичне заняття №6</b> Конструювання меню. Стандартні меню. Контекстні меню. Робота з дизайнером меню Menu Designer	2
7.	<b>Практичне заняття №6</b> Графіка. Клас TCanvas	2

### 7. Теми лабораторних занять

№ зан.	Назва теми	Кількість годин
1.	<b>Лабораторне заняття № 1</b> Класи і об'єкти в с ++	2
2.	<b>Лабораторне заняття № 2</b> Перевантаження операцій	2
3.	<b>Лабораторне заняття № 3</b> Потокові класи	2
4.	<b>Лабораторне заняття №4</b> Розробка додатків. Послідовність розробки та виконання	2

№ зан.	Назва теми	Кількість годин
	додатків	
5.	<b>Лабораторне заняття №5</b> Робота зі списками. Організація багатосторінкових списків. Обробка текстових файлів	2
6.	<b>Лабораторне заняття №6</b> Використання швидких кнопок. Динамічне створення масиву кнопок	2
7.	<b>Лабораторне заняття №7</b> Організація управління додатком. Диспетчеризація дій. Створення інструментальних панелей	2

### 8. Самостійна робота

№ зан.	Назва теми	Кількість годин
1	<b>Тема: 1.1</b> Основні етапи розв'язку задач з використанням комп'ютера. Життєвий цикл програм. Постановка задачі для розв'язування на комп'ютері. Поняття алгоритму. Основні способи запису алгоритмів: блок - схеми, алгоритмічні мови. Класифікація алгоритмічних мов. . [4] с. 5 - 175	4
2	<b>Тема: 1.2</b> Складений оператор. [2] с.4 8 - 218	2
3	<b>Тема: 2.3</b> Виконання операцій з рядками. [2] с. 125 - 330	4
4	<b>Тема: 2.1</b> Використання динамічної пам'яті при розробці програм на мові C ++. [1] с. 71 - 275	4
5	<b>Тема: 2.2</b> Перетворення типів даних.. [3] с. 223-328	4
7	<b>Тема: 3.1</b> Використання графічних функцій.. [3] с. 207 - 320	4
8	<b>Тема: 3. 2.</b> Структура вибору, оператор варіанту. [3] с.1 9 - 196	4
	<b>Всього</b>	<b>26</b>

## 9. Індивідуальні завдання

Не передбачено навчальним планом

## 10. Методи навчання

I. По джерелу знань:

- 1.1. Словесні – лекція;
- 1.2. Діалогічні (бесіда, диспут, семінар);
- 1.3. Практичні (вправи, практичні завдання);

II. По пізнавальній діяльності:

- 2.1. Репродуктивний (відтворювальний) – розв’язання задач;
- 2.2. Частково-пошуковий (самостійна робота),
- 2.3. Евристичний (проблемно-розвиваючий) – евристична бесіда, евристичне фронтальне опитування;
- 2.4. Метод проблемного викладання;

III. Методи по логіці навчання:

- 1.1. Індуктивний – від часткового до загального;
- 1.2. Дедуктивний – від загального до часткового;
- 1.3. Синтез;
- 1.4. Аналіз;
- 1.5. Порівняння.

IV. Методи колективної розумової діяльності:

- 4.1. Пізнавальна суперечка;
- 4.2. Навчальні дискусії;
- 4.3. Метод „мозкового штурму”;

V. Імітаційні методи навчання (ділова гра).

## 11. Методи контролю

Види контролю	Застосування
Поточний контроль	Атестація, модульний контроль знань, контроль виконання самостійної роботи студентів
Підсумковий контроль	Семестровий контроль (залік)



## 12. Розподіл балів, які отримують студенти

Підсумковий контроль знань студентів з дисципліни «Об'єктно-орієнтоване програмування» проводиться у формі екзамену.

Згідно з вимогами регламенту навчального процесу в коледжі, викладачі зобов'язані контролювати відвідування студентами лекцій, семінарських та лабораторних занять, здійснювати систематичний поточний контроль знань студентів протягом всього строку вивчення дисципліни. За результатами такого контролю студенти допускаються або не допускаються до складання екзамену.

Допуск до екзамену студент отримує від викладача, який протягом семестру вів у групі лекції, лабораторні заняття з цієї дисципліни. Студенти, які мали заборгованість (пропуски занять, незадовільні оцінки, невиконаний курсовий проект) і не ліквідували її до початку сесії, до заліку не допускаються.

**1. "Відмінно"** - студент дає обґрунтовані, глибокі й теоретично правильні відповіді на поставлені питання; правильно і раціонально виконує практичні завдання, обґрунтовуючи їх вичерпними поясненнями; робить висновки та узагальнення; використовує сучасні прилади для виконання поставленого завдання; вміло виконує лабораторні роботи, виконує складання програм будь якого рівня, не допускаючи помилок, створює програмні засоби, використовуючи оптимальні алгоритми.

**2. "Добре"** - студент володіє знаннями матеріалу на рівні вимог, наведених вище, але у розкритті питань ним були допущені незначні помилки у термінах; незначні неточності у виконаних практично завданнях, що не суттєво впливають на розрахунки і роботу програмних засобів.

**3. "Задовільно"** - студент допускає значні помилки при відповіді на теоретичні запитання: при виконанні практичних завдань необхідні незначні підказки з боку вчителя, але отримані програмні продукти являються працездатними і виконують базові функції;

**4. "Незадовільно"** - студент допускає неправильні відповіді або зовсім не відповідає на теоретичні запитання; не виконує практичних завдань; не вміє підібрати інструмент рішення поставлених завдань.

## 13. Методичне забезпечення

1. Методичні вказівки до самостійної роботи студентів з предмету «Об'єктно-орієнтоване програмування»;
2. Методичні матеріали до поточного і підсумкового контролю:
  - завдання до проведення тематичного контролю знань;
  - завдання до проведення контрольних робіт;
  - завдання та методичні вказівки до курсового проектування;
  - питання до екзамену;

## 14. Рекомендована література

## 14. Рекомендована література

### Основна

1. Об'єктно-орієнтоване програмування мовою Java. Навчальний посібник. – Вінниця: ВНТУ, 2016. – 101 с.
2. Об'єктно-орієнтоване програмування : підручник. У 2-х ч. Ч. 2. Об'єктно-орієнтований підхід до розробки програмного забезпечення / С. М. Алхімова. - Київ: КГП ім. Ігоря Сікорського, Вид-во «Політехніка», 2019.- 192 с
3. Програмування на Java 8: Навчальний посібник для студентів факультету комп'ютерних наук та кібернетики. – К.: ЛОГОС, 2017. – 186 с.

### Додаткова

4. ООП Конспект лекцій –2022р 79с.
5. Муляр В. П. Основи розробки додатків з використанням технології JavaFX. Комп'ютерно-інтегровані технології: освіта, наука, виробництво. 2018. Вип. № 30-31. С. 104–110.
6. Муляр В. П. Розробка JavaFX-додатків із використанням Scene Builder. Комп'ютерно-інтегровані технології: освіта, наука, виробництво. 2020. Вип. № 39. С. 181–189.
7. Муляр В. П., Яцюк С. М. Елементи комп'ютерної графіки у візуалізації результатів моделювання фізичних явищ і процесів. Комп'ютерно-орієнтовані технології: освіта, наука, виробництво. 2016. № 23. С. 80–84.
8. Спирінцева О. В., Литвинов О. А., Герасимов В. В. Java-технології та мобільні пристрої. Алгоритми і структури даних: навч. посіб. Д.: Вид-во ДНУ ім. О. Гончара, 2016. 140 с.

### Інформаційні ресурси

1. Java Tutorial. URL: <https://www.w3schools.com/java/default.asp>  
Java. Классы. Объектно-ориентированное программирование. URL: <https://metanit.com/java/tutorial/3.1.php>
2. Підручник з Java. URL: <https://www.javatpoint.com/java-tutorial>
3. Apache NetBeans. URL: <https://netbeans.apache.org/download/index.html>
4. Java Course. URL: <http://java-course.ru/begin/introduce/>
5. Java SE Downloads. URL: <https://www.oracle.com/java/technologies/javase-downloads.html>
6. JavaFX. URL: <https://gluonhq.com/products/javafx/>
7. Scene Builder. URL: <https://gluonhq.com/products/scene-builder/>