

Міністерство освіти і науки України

Національний аерокосмічний університет ім. М.Є. Жуковського
«Харківський авіаційний інститут»

Кафедра інформаційних технологій проектування (№ 105)

ЗАТВЕРДЖУЮ

Голова НМК 2



Д.М.Крицький
(ініціали та прізвище)

« 31 » 08 2021 р.

**РОБОЧА ПРОГРАМА ОБОВ'ЯЗКОВОЇ
НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

Об'єктно-орієнтоване програмування
(назва навчальної дисципліни)

Галузь знань: 12 "Інформаційні технології"
(шифр і найменування галузі знань)

Спеціальність: 126 "Інформаційні системи та технології"
(код та найменування спеціальності)

Освітня програма: Інформаційні системи та технології підтримки
віртуальних середовищ

(найменування освітньої програми)

Форма навчання: денна

Рівень вищої освіти: перший (бакалаврський)

Харків 2021 рік

Робоча програма Об'єктно-орієнтоване програмування
(назва дисципліни)
для студентів за спеціальностями 126 "Інформаційні системи та технології"
«08» 06 2021 р., - 13 с.

Розробник: Овсяннік В.М., доцент к.т.н.,
(прізвище та ініціали, посада, наукова ступінь та вчене звання)  (підпис)

Робочу програму розглянуто на засіданні кафедри інформаційних технологій
проектування
(назва кафедри)

Протокол № 1 від «31» серпня 2021 р.

Завідувач кафедри к.т.н., доцент
(науковий ступінь та вчене звання)  (підпис) Д.М.Крицький
(ініціали та прізвище)

1. Опис навчальної дисципліни

Найменування показника	Галузь знань, спеціальність, освітня програма, рівень вищої освіти	Характеристика навчальної дисципліни (денна форма навчання)
Кількість кредитів – 5	Галузь знань <u>12 «Інформаційні технології»</u> (шифр і назва)	Цикл професійної підготовки
Кількість модулів – 2	Спеціальність <u>126 “Інформаційні системи та технології”</u> (шифр і назва)	Навчальний рік:
Кількість змістових модулів – 4		2021 / 2022
Індивідуальне завдання – <u>Розрахунково-графічна робота</u>		Семестр
Загальна кількість годин – 64/150		3-й (1-й скор.ф.навч.)
Кількість тижневих годин для денної форми навчання: аудиторних – 4 самостійної роботи студента – 5,4		Рівень вищої освіти: перший (бакалаврський)
		Практичні, семінарські 0 годин
		Лабораторні¹⁾ 32 години
		Самостійна робота 86 годин
		Вид контролю модульний контроль, іспит

Співвідношення кількості годин аудиторних занять до самостійної роботи становить для денної форми $64/86=0.74$.

¹⁾ Аудиторне навантаження може бути зменшене або збільшене на одну годину в залежності від розкладу занять.

2. Мета і завдання навчальної дисципліни

Мета вивчення навчальної дисципліни полягає у підготовці спеціалістів з інформаційних систем до виконання робіт з розроблення програмного забезпечення з використанням принципів та методів об'єктно-орієнтованого програмування.

Основними завданнями дисципліни є вивчення засобів розробки програмного забезпечення для роботи під керівництвом ОС Windows з широким використанням можливостей об'єктно-орієнтованого програмування, правила будування програмних засобів в середовищах візуального програмування, відлагодження програм та розв'язання типових задач.

Згідно з вимогами освітньо-професійної програми студенти повинні досягти таких загальних та фахових компетентностей:

ЗК 2. Здатність застосовувати знання у практичних ситуаціях.

ЗК 3. Здатність до розуміння предметної області та професійної діяльності.

ЗК 4. Здатність спілкуватися іноземною мовою.

КЗ 6. Здатність до пошуку, оброблення та узагальнення інформації з різних джерел.

КС 2. Здатність застосовувати стандарти в області інформаційних систем та технологій при розробці функціональних профілів, побудові та інтеграції систем, продуктів, сервісів і елементів інфраструктури організації.

Програмні результати навчання:

ПР 10. Розуміти і враховувати соціальні, екологічні, етичні, економічні аспекти, вимоги охорони праці, виробничої санітарії, пожежної безпеки та існуючих державних і закордонних стандартів під час формування технічних завдань та рішень.

У результаті вивчення навчальної дисципліни студент повинен

знати:

- об'єктно-орієнтовану мову програмування C++;
- найбільш поширені інструментальні засоби для програмування у середовищі Microsoft Visual Studio;
- особливості роботи програм під керуванням Windows;
- правила будування програм в візуальних середовищах програмування;
- алгоритми вирішення типових задач САПР;

уміти:

- застосовувати об'єктно-орієнтований підхід під час проектування складних програмних систем;
- проектувати компоненти програмного забезпечення;
- проектувати людино-машинний інтерфейс інформаційних систем;
- реалізовувати прототипи архітектури програмного забезпечення;
- установлювати, налаштовувати та обслуговувати системне, інструментальне і прикладне програмне забезпечення та інформаційні системи.

Студент має надбати **навички** самостійного проектування, розробки та відлагодження застосувань, призначених для розв'язання типових математичних та прикладних задач, що виникають під час розробки функціональних складових модулів САПР.

Міждисциплінарні зв'язки: навчальна дисципліна розрахована на студентів, що мають підготовку по інформатиці і програмуванню, знання основних базових алгоритмів і базових конструкцій мов програмування. Знання та уміння можуть бути використані для розробки алгоритмів і програм на базі об'єктно-орієнтованого підходу. Викладання дисципліни базується на курсах «Основи програмування», «Алгоритмізація і програмування» та «Дискретна математика». Знання та навички, що здобувають студенти під час вивчення дисципліни, використовуються при вивченні таких дисциплін, як «Технологія розробки програмних продуктів», «Інструментальні засоби візуального програмування», «Організація баз даних та знань»

3. Програма навчальної дисципліни

Модуль 1. Об'єктно-орієнтований аналіз та технологія ООП

Змістовий модуль 1. Об'єктно-орієнтований аналіз і проектування

Тема 1. Поняття об'єктно-орієнтованого аналізу, проектування та програмування.

Поняття об'єкта, сутності, концепту, атрибуту, відношення, класу, предметної області простору проблем та простору рішень. Визначення цих понять та їх використання. Об'єкт як абстракція реального світу і понятійна структура, його поведінка, обумовлена властивостями даного об'єкта і його зв'язками з іншими об'єктами.

Тема 2. Об'єктна модель предметного середовища, принципи її побудови

Опис предметного середовища та постановка задачі. Принципи побудови моделі, їх використання та приклади застосування.

Тема 3. Поняття об'єктів і класів та їх взаємовідносин.

Поняття класи і об'єктів, формат їх опису та складові частини. Опис класу і створення об'єкта. Визначення властивостей класу. Псевдо-змінна this. Визначення методів класу.

Змістовий модуль 2. Технологія об'єктно-орієнтованого програмування

Тема 4. Основи об'єктно-орієнтованого проектування мовою UML.

Поняття та призначення UML як мови, що призначена для аналізу, візуального моделювання та проектування програмного забезпечення. Реалізація в UML таких загальних функцій як візуалізація, специфікація, конструювання та документування елементів програмних систем. Типи моделей UML та їх використання, приклади реалізації. Структурні діаграми, діаграми поведінки та взаємодії. Позначення в термінах нотації Буча. Переведення діаграм класів в C++.

Тема 5. Основи об'єктно-орієнтованої мови програмування.

Коротка історична довідка стосовно мов Сімула та Smalltalk. Об'єктно-орієнтоване програмування як метод програмування, заснований на поданні програми у вигляді сукупності взаємодіючих об'єктів, кожен з яких є екземпляром певного класу, а класи є членами певної ієрархії наслідування. ООП як одна з парадигм програмування, яка розглядає програму як множину «об'єктів», що взаємодіють між собою. Три основні концепції ООП: інкапсуляція, успадкування та поліморфізм.

Тема 6. Абстрагування даних та інкапсуляція.

Класифікація операцій, їх пріоритети, правила використання і приклади застосування. Обчислення виражень.

МОДУЛЬ 2. Розроблення Windows-застосунків.

Змістовий модуль 3. Перевантаження операцій і бібліотеки класів

Тема 7. Конструктори та деструктори класів.

Призначення і формат опису конструкторів і деструкторів. Особливості виклику конструкторів і деструкторів класів. Розміщення екземплярів класів у «купі». Огляд CRest як прикладу класу, що має декілька конструкторів, а також інші компоненти. Використання вказівника this.

Тема 8. Перевантаження операцій та функцій.

Перевантаження унарних і бінарних операторів. Перевантаження операцій потокового введення – виведення. Використання віртуальних функцій.

Тема 9. Статичні, константні члени класів, дружні функції та класи.

Призначення статичних членів класів, правила їх опису та використання, особливі властивості. Константні об'єкти, константні функції, змінювані члени константних об'єктів (mutable). Приклади опису та використання константних членів класів.

Тема 10. Композиція та колекція об'єктів.

Поняття композиція (інтеграція) об'єктів та їх залежність від життєвого циклу програмної системи. Способи зв'язку та взаємодії об'єктів. Контейнери, ітератори і алгоритми стандартної бібліотеки STL.

Змістовий модуль 4. Успадкування, поліморфізм та шаблони класів

Тема 11. Просте та множинне успадкування.

Успадкування. Відкриті і закриті похідні класи. Ієрархія і композиція. Правила доступу. Множинне успадкування. Порядок виклику конструкторів.

Тема 12. Реалізація поліморфізму.

Поліморфізм як концепція програмування, відповідно до якої використовується спільний інтерфейс для обробки даних різних спеціалізованих типів. Реалізація поліморфізму шляхом використання віртуальних функцій та перевантаження операцій. Переваги використання поліморфізму. Приклади ієрархій класів, що використовують поліморфізм.

Тема 13. Шаблони функцій і класів.

Шаблони як засіб мови C++, який призначений для кодування узагальнених алгоритмів, без прив'язки до таких параметрів, як тип даних та розмір буфера. Визначення та призначення шаблонів функцій та шаблонів класів, їх опис та правила використання. Параметри шаблонів, принципи роботи шаблонних функцій. Ініціалізація шаблонного класу. Склад бібліотеки стандартних шаблонів (STL).

4. Структура навчальної дисципліни

Назви змістових модулів і тем	Кількість годин					
	денна форма					
	усього	у тому числі				
л		п	лаб	курс	с.р.	
1	2	3	4	5	6	7
Модуль 1. Об'єктно-орієнтований аналіз та технологія ООП						
Змістовий модуль 1. Об'єктно-орієнтований аналіз і проектування						
Тема 1. Поняття об'єктно-орієнтованого аналізу, проектування та програмування	2	2	—	—	—	—
Тема 2. Об'єктна модель предметного середовища, принципи її побудови	12	2	—	4	—	6
Тема 3. Поняття об'єктів і класів та їх взаємовідносин.	16	4	—	4	—	6

Разом за змістовим модулем 1	30	8	—	8	—	14
Змістовий модуль 2. Технологія об'єктно-орієнтованого програмування						
Тема 4. Основи об'єктно-орієнтованого проектування мовою UML	10	4	—	2	—	4
Тема 5. Основи об'єктно-орієнтованої мови програмування	10	2	—	2	—	6
Тема 6. Абстрагування даних та інкапсуляція	20	2	—	4	—	16
Разом за змістовим модулем 2	40	8	—	8	—	26
Усього годин за модулем 1	70	16	—	16	—	40
Модуль 2. Розроблення Windows-застосунків						
Змістовий модуль 3. Перевантаження операцій і бібліотеки класів						
Тема 7. Конструктори та деструктори класів	10	2	—	2	—	6
Тема 8. Перевантаження операцій та функцій	4	2	—	2	—	—
Тема 9. Статичні, константні члени класів, дружні функції та класи	9	1	—	2	—	6
Тема 10. Композиція та колекція об'єктів	17	1	—	2	—	14
Разом за змістовим модулем 3	40	6	—	8	—	26
Змістовий модуль 4. Успадкування, поліморфізм та шаблони класів						
Тема 11. Просте та множинне успадкування	14	2	—	4	—	8
Тема 12. Реалізація поліморфізму	12	4	—	4	—	4
Тема 13. Шаблони функцій і класів	12	4	—	—	—	8
Разом за змістовим модулем 4	38	10	—	8	—	20
Усього годин за модулем 2	65	16	—	16	—	46
Усього годин за дисципліною	150	32	—	32	—	86

5. Теми семінарських занять

Семінарські заняття навчальним планом не передбачені.

6. Теми практичних занять

Практичні заняття навчальним планом не передбачені.

7. Теми лабораторних занять

№ з/п	Назва теми	Кількість годин
1	Розроблення застосування з використанням наданого класу	4
2	Розроблення додатків з використанням базових елементів ООП	4
3	Застосування спадкування	4
4	Розроблення додатків з використанням шаблонів функцій і класів	4
5	Математичні обчислення з обробленням виняткових ситуацій	4
6	Програмування застосунків на основі класу діалогового вікна	4
7	Використання основних бібліотек Microsoft .NET Framework	4
8	Використання елементів управління у додатках з графічним інтерфейсом користувача	4
	Разом	32

8. Індивідуальне завдання – розрахунково-графічна робота

8.1 Зміст та графік виконання розрахунково-графічної роботи

№	Етап	Тиждень семестра	Форма звіту	% об'єму
1	Збір матеріалу по заданій темі	4	Лист затвердження завдання	5
2	Читання даних з файлу у внутрішні структури програми	6	Розділ записки (електронна форма) і блок програми	30
3	Вивід даних на монітор	7	Блок програми	50
4	Сортування даних по заданому полю	9	Блок програми	80
5	Оформлення пояснювальної записки і завершення розробки програми	11	Пояснювальна записка	90
6	Здача пояснювальної записки і програми	12	Записка і програма	100

Етап 1. Вибір та затвердження теми у викладача. У кожному завданні повинна бути обрана оригінальна тема і зібраний матеріал по ній. Матеріал по темі повинен бути оформлений у вигляді таблиці даних, що має не менше 25 рядків і не менше 7 стовпців. Іншими словами, один рядок таблиці повинен являти собою структуру (struct), що складається з компонентів різних типів і містить конкретні дані. Наприклад, наведена вище таблиця з графіком виконання домашнього завдання містить 5 полів даних, два з яких мають строковий тип, а три – цілочисельний.

Типи полів даних з обраної тематики повинні бути різними і вибиратися з наступного переліку, який, втім, Ви можете розширити:

- bool;
- int, long int, char, __int64, unsigned int і т.п. .;
- char, unsigned char;
- float, double;
- char *, char [];
- enum;
- покажчики;
- структури;
- масиви.

Перший етап виконання домашнього завдання повинен завершитися вибором теми і полів даних, тобто власне описом структури даних, які повинен затвердити викладач. Наповнення таблиці даних конкретних даних на цьому етапі не обов'язково.

Етап 2. На цьому етапі необхідно зібрати таблицю даних в звичайному текстовому файлі і написати програмний код, що забезпечує читання цих даних з файлу у внутрішні структури даних програми. Попередньо потрібно вирішити, що собою будуть представляти ці самі «внутрішні структури даних», тобто їх треба спроектувати. Такий внутрішньою структурою програми може бути масив структур (struct) або класів.

Етап 3. На цьому етапі необхідно вирішити задачу представлення даних вибагливому, але справедливому, користувачеві в максимально зручному для нього вигляді. Якщо додаток реалізовано у вигляді консольного, то зручний формат - це висновок одного рядка таблиці в

одному рядку екрану. У цьому випадку програма повинна управляти розмірами вікна консольного застосування, дисплейного буфера, числом символів в рядку і числом рядків, колірними атрибутами символів. Ці настройки програма повинна читати з ini-файлу під час запуску і використовувати їх для відображення даних.

Етап 4. Обробка даних, яка полягає у виконанні наступних дій:

- сортування даних по заданому користувачем полю (компоненту структури);
- вибірці екстремальних значень із заданого (числового) поля;
- обчислення середнього значення для заданого (числового) поля.

Природно, що результати обробки повинні відобразитися на моніторі, а їх запис в файл не є обов'язковим.

Етап 5. На цьому етапі необхідно завершити оформлення пояснювальної записки відповідно до вимог, викладених нижче в розділі «Вимоги до оформлення пояснювальної записки».

Етап 6. На цьому заключному етапі необхідно здати пояснювальну записку викладачеві на перевірку, при необхідності виправити зауваження і, на завершення, переконати викладача в тому, що Ваша програма працює і не містить помилок.

9. Самостійна робота

Для опанування матеріалу дисципліни "Об'єктно-орієнтоване програмування" окрім лекційних та лабораторних занять, тобто аудиторної роботи, значну увагу необхідно приділяти самостійній роботі.

Основні форми самостійної роботи студента:

1. Вивчення лекційного матеріалу.
2. Вивчення окремих тем або питань, що передбачені для самостійного опрацювання.
3. Вивчення основних термінів та понять за темами дисципліни.
4. Підготовка до лабораторних занять.
5. Контрольна перевірка кожним студентом знань за питаннями для самодіагностики.
6. Підготовка до проміжного та підсумкового модульного контролю.
7. Систематизація вивченого матеріалу перед іспитом.
8. Оформлення звітів з лабораторних робіт.
9. Робота з опрацювання та вивчення рекомендованої літератури.
10. Виконання індивідуальної розрахунково-графічної роботи.

Теми і запитання для самостійного опрацювання

№ з/п	Назва теми	Кількість годин
1	<u>Об'єктна модель предметного середовища, принципи її побудови.</u> Ознаки складних програмних систем. Переваги об'єктно-орієнтованого підходу до розроблення програмних систем. Відношення між об'єктами. Відношення між класами. Прості і складні програмні системи. Поняття про "покоління" об'єктів.	10
2	<u>Основи об'єктно-орієнтованого проектування мовою UML.</u> Історія розвитку мови UML. Види UML-діаграм. CASE-засоби. Основні поняття візуального моделювання. Графічні нотації моделювання, які широко використовуються. Основні розробники мови UML. Види відносин на діаграмі класів	10

3	<u>Абстрагування даних та інкапсуляція.</u> Призначення та використання структур.	10
4	<u>Статичні, константні члени класів, дружні функції та класи.</u> Синтаксис опису класу. Особливості статичних елементів класу. Специфікатори доступу до елементів класу. Порядок ініціалізації об'єкта класу. Призначення та використання посилання this. Абстрактні класи та віртуальні функції. Пізніє та раннє зв'язування.	9
5	Шаблони функцій і класів	6
6	<u>Оброблення виняткових ситуацій.</u> Призначення та використання елементів класу CException. Використання вкладених блоків try. Розробка власних класів виключень. Генерування користувальницьких виключень. Перехоплення користувальницьких виключень.	6
7	<u>Стандартні бібліотеки класів середовищ розробника програм.</u> Способи використання директиви using. Призначення основних просторів імен бібліотеки класів платформи Microsoft .NET. Налаштування програм у середовищі Microsoft Visual Studio	10
8	<u>Бібліотеки класів реалізації функціональних можливостей Windows.</u> Використання засобів SDK щодо розроблення програм. Перетворення типів, які визначаються користувачем. Особливості потоків введення-виведення даних, що використовують буферизацію.	6
9	<u>Розробка графічних інтерфейсів користувача (меню, панелі інструментів, шаблони діалогових вікон тощо).</u> Призначення та застосування класу Application. Функціональні можливості класу Control. Використання "колекції" візуальних елементів управління форми. Застосування оброблювачів подій рівня форми. Розміщення візуальних елементів управління на формі. Ієрархія класів-предків класу Form та його функціональні можливості. Основні властивості та події класу Control. Склад та призначення основних файлів проекту типу Windows Application у середовищі Visual Studio.	10
10	<u>Основи програмування, керованого подіями.</u> Інтерфейси зворотного виклику. Реалізація групових викликів. Реєстрація подій у середовищі Visual Studio	2
11	<u>Обробники подій від миші, клавіатури, команд меню, елементів управління тощо.</u> Обробка подій пунктів меню. Обробка подій кнопок панелі інструментів. Акселератори та підказки, які спливають. Властивості та використання компонента Button. Властивості та використання компонента TextBox. Властивості та використання компонента StaticText. Властивості та використання компонента GroupBox. Властивості та використання компонента RadioButton. Властивості та використання компонента CheckBox.	5
12	<u>Розроблення DLL-бібліотек.</u> Проблема "Аду DLL" та напрями її вирішення.	2
	Разом	86

10. Методи навчання

Лекції проводяться з використанням основних розділів конспекту лекцій в електронній формі, елементів мультимедійної підтримки курсу (відео фрагментів), демонстрацій окремих прийомів роботи з інструментальним середовищем та/або роздаткових матеріалів у вигляді схем та діаграм.

Лабораторні роботи виконуються з використанням навчальних (демонстраційних) та ліцензованих робочих версій сучасного програмного забезпечення.

Самостійна робота включає підготовку до лабораторних робіт, модульного контролю та іспиту, виконання поза-аудиторної частини індивідуального завдання і вивчення вказаних вище тем за конспектом, літературними джерелами та програмною документацією.

11. Методи контролю

Поточний контроль здійснюється відповідно до повноти, своєчасності та якості виконання лабораторних робіт та домашнього завдання згідно з “Інструкцією з модульно-рейтингового оцінювання знань студентів з дисципліни “Об’єктно-орієнтованого програмування”; проміжний (модульний) контроль – письмові контрольні роботи на 8-му та 16-му тижнях; підсумковий контроль – письмовий іспит.

12. Критерії оцінювання та розподіл балів, які отримують студенти

12.1. Розподіл балів, які отримують студенти (кількісні критерії оцінювання)

Складові навчальної роботи	Бали за одне заняття (завдання)	Кількість занять (завдань)	Сумарна кількість балів
Змістовний модуль 1			
Робота на лекціях	0...1	5	0...5
Виконання і захист лабораторних (практичних) робіт	3...5	4	12...20
Модульний контроль	10...15	1	10...15
Змістовний модуль 2			
Робота на лекціях	0...1	5	0...5
Виконання і захист лабораторних (практичних) робіт	3...5	4	12...20
Модульний контроль	10...15	1	10...15
Виконання і захист РГР (РР, РК)	16...20	1	16...20
Усього за семестр			60...100

Семестровий контроль (іспит/залік) проводиться у разі відмови студента від балів поточного тестування та за наявності допуску до іспиту/заліку. При складанні семестрового іспиту/заліку студент має можливість отримати максимум 100 балів.

Білет для іспиту складається з двох теоретичних питань та одного практичного завдання. Максимальна кількість балів за кожне питання - 30 балів, за практичне завдання – 40 балів, загальна сума – 100 балів.

12.2. Якісні критерії оцінювання

Необхідний обсяг знань для одержання позитивної оцінки:

- мати уявлення про весь теоретичний матеріал дисципліни;
- знати в повному обсязі не менш половини тем теоретичного матеріалу;
- знати додатковий матеріал по трьом темам.

Необхідний обсяг умінь для одержання позитивної оцінки:

- володіти технологією застосування теоретичних знань на практиці;
- уміти самостійно знаходити довідкові відомості;
- уміти застосовувати на практиці отримані знання.

12.3 Критерії оцінювання роботи студента протягом семестру

Задовільно (60-74). Знати основні положення теоретичного матеріалу. Вміти користуватися засобами інструментального середовища Visual Studio. Вміти розробляти додатки Windows, що вирішують задачі з використанням таких типів даних, як структури та простіші класи.

Добре (75-89). Знати основний теоретичний матеріал в повному обсязі. Володіти технологією пошуку довідкової літератури. Вміти розробляти та налагоджувати додатки в інструментальному середовищу Visual Studio, що опрацьовують успадковані класи.

Відмінно (90-100). Знати основний і додатковий теоретичний матеріал в повному обсязі. Добре орієнтуватися в довідковій літературі. Вміти вирішувати задачі та розробляти додатки з використанням віртуальних функцій та шаблонних класів. Вміти розробляти додатки, що використовують графічний режим роботи.

Переведення підсумкових семестрових оцінок в оцінки за національною шкалою та в оцінки за міжнародною шкалою ECTS здійснюється відповідно до «Положення про модульно-рейтингову систему оцінювання знань студентів» за наведеною нижче таблицею.

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою
90–100	A	відмінно
83–89	B	добре
75–82	C	
68–74	D	задовільно
60–67	E	
0–59	FX	незадовільно

13. Методичне забезпечення

Завдання та сценарії виконання лабораторних робіт у електронній формі та допоміжні приклади застосувань Windows, що надаються студентам на сервері кафедри.

14. Рекомендована література

14.1. Базова

1. Кравець П.О. Об'єктно-орієнтоване програмування: навч. посібник/ П.О.Кравець. – Львів: Видавництво Львівської політехніки, 2012. – 624с.
2. Юрченко І.В., Сікора В.С. Інформатика та програмування. Частина 2.– Чернівці: Видавець Яворський С.Н., 2015.– 210 с.
3. Г.Шилдт. С++: базовый курс / Пер. с англ. – М.: "Диалектика Вильямс", 2018. – 624с..
4. Буч Г., Рамбо Д., Якобсон И. Язык UML. Руководство пользователя. 2-е изд./Пер. с англ. Мухин Н. – М.: ДМК Пресс, 2015.-496с.:ил.
5. Овсянник В.Н. Язык С++ не для чайников. Учебное пособие – [Електронний ресурс] – Режим доступа: – 10.1.108.193.(01.09.2017)

14.2. Допоміжна

6. Дьюхерст С. С++. Священные знания, 2-е издание. – Пер. с англ. – СПб.: Символ-Плюс, 2016. – 240с.
7. Э.Гамма, Р.Хелм, Р.Джонсон, Д.Влиссидес. Приемы объектно-ориентированного проектирования. Паттерны проектирования/Пер. с англ. – СПб.: "Питер", 2018. – с.368.
8. Орлов С.А. Технологии разработки программного обеспечения: Учебник. – СПб.: Питер, 2012. – 464 с.: ил. ISBN 5-94723-145-Х
9. Эрих Гамма, Ричард Хелм, Ральф Джонсон, Джон Влиссидес. Приемы объектно-ориентированного проектирования. Паттерны проектирования. Издательство: "Питер", 2015. - 368 с.
10. Джеффри Рихтер. Программирование на платформе Microsoft .NET Framework / Пер. с англ. – М.:Издательско-торговый дом «Русская редакция», 2014 - 512с.
11. Дон Бокс, Крис Селлз. Основы платформы .NET, том 1. Общезыковаемая исполняющая среда /Пер. с англ. – М.; СПб.: К.: Издательский дом "Вильямс", 2015. – с.288. ISBN 5-8459-0455-2

14.3. Інформаційні ресурси

12. База знань Microsoft Developer Network (MSDN) – [Електронний ресурс] – Режим доступу: – <http://msdn.microsoft.com/> (01.06.2017)
13. Офіційний сайт компанії Microsoft щодо технологій WPF та Windows Forms – [Електронний ресурс] – Режим доступу: – <http://window-sclient.net> (01.06.2017)
14. Internet-інститут інформаційних технологій – [Електронний ресурс] – Режим доступу: – www.intuit.ru (01.06.2015)