

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ**  
**Волинський національний університет імені Лесі Українки**  
**Факультет інформаційних технологій і математики**  
**Кафедра комп'ютерних наук та кібербезпеки**

**СИЛАБУС**

**нормативного освітнього компонента**

**ПРОГРАМУВАННЯ**

**підготовки бакалавра**

**спеціальності 122 Комп'ютерні науки**

**освітньо-професійної програми**

**Комп'ютерні науки та інформаційні технології**

Луцьк – 2022

**Силабус навчальної дисципліни «Програмування»** підготовки бакалавра, галузі знань 12 Інформаційні технології, спеціальності 122 Комп'ютерні науки, за освітньою програмою Комп'ютерні науки та інформаційні технології

**Розробник:** Глинчук Л. Я., доцент кафедри комп'ютерних наук та кібербезпеки, кандидат фізико-математичних наук.

**Погоджено**

Гарант освітньо-професійної програми:



Гришанович Т. О.

**Силабус освітнього компонента затверджено на засіданні кафедри комп'ютерних наук та кібербезпеки**

протокол № 2 від 29 вересня 2022 р.

Завідувач кафедри:



Гришанович Т. О.

## I. Опис навчальної дисципліни

Найменування показників	Галузь знань, спеціальність, освітньо-професійна /освітньо-наукова/освітньо-творча програма, освітній рівень	Характеристика освітнього компонента
Денна форма навчання	<i>12 Інформаційні технології</i>  <i>122 Комп'ютерні науки</i>  <i>Комп'ютерні науки та інформаційні технології</i>  <i>Бакалавр</i>	<b>Нормативний</b>
Кількість годин/кредитів 300/10		<b>Рік навчання: II</b>
		<b>Семестр: 3, 4</b>
		<b>Лекції: 34 год/24 год.</b>
		<b>Лабораторні роботи: 54 год/34 год.</b>
		<b>Самостійна робота: 138 год.</b>
ІНДЗ: немає		<b>Консультації: 8 год/6 год.</b>
	<b>Форма контролю: залік, екзамен</b>	
<b>Мова навчання</b>	<i>Українська</i>	

## II. Інформація про викладача

ППП Глинчук Людмила Ярославівна

Науковий ступінь кандидат фізико-математичних наук

Вчене звання -

Посада доцент кафедри комп'ютерних наук та кібербезпеки

Контактна інформація тел.:095 890 42 46, e-mail: Hlynchuk.Ludmila@vnu.edu.ua

Дні занять <http://194.44.187.20/cgi-bin/timetable.cgi?n=700>

## III. Опис дисципліни

**1. Анотація курсу.** Дисципліна «ПРОГРАМУВАННЯ» належить до переліку нормативних навчальних дисциплін програми підготовки бакалавра. Спрямована на підвищення рівня формування у студентів знань та умінь, які дадуть теоретичний і практичний фундамент, необхідний для вивчення загальних принципів програмування, базових термінологій. Оволодіння навичками програмування класичних задач, алгоритмів їх розв'язання та технологією розробки програм на мові програмування C++, C# з використанням консольних та візуальних середовищ.

**2. Предреквізити.** Базові знання з дисциплін: «Алгоритми та структури даних», «Програмування I курс», «Вибрані питання теоретичної інформатики та інформаційних технологій I курс», «Дискретна математика».

**Постреквізити.** Знання та вміння, набуті в результаті вивчення дисципліни, можна використати у подальшому навчанні та у професійній діяльності. А також, для написання індивідуальних, курсових та випускних кваліфікаційних робіт (бакалаврської, магістерської).

### 3. Мета і завдання навчальної дисципліни.

**Метою** викладання навчальної дисципліни «Програмування» є засвоєння студентами основних концепцій, принципів та понять сучасного програмування, що створюють основу теоретичних досліджень і практичних розробок в області універсальних мов програмування (C++, C#). Вивчення сучасного візуального (кросплатформного) програмування; розуміння поняття віджету (компоненту), події, слота, сигналу та особливості роботи з ними; розробка графічного інтерфейсу додатків з використанням інструментів середовищ Visual Studio (C# Windows Forms) та Qt.

**Завдання:** основними завданнями вивчення дисципліни “Програмування” у III та IV семестрах є подальше вивчення ООП, познайомитися з можливостями використання графічних бібліотек, розглянути особливості візуального програмування та дати наступні **знання:** техніку застосування дружніх класів, методів та функцій, шаблонів функцій та шаблонів класів, перевантаження шаблонів для різних випадків; техніку обробки помилок та виняткових ситуацій; техніку програмування графічних зображень; техніку програмування додатків в середовищі Visual Studio з використанням C#, в середовищі Qt мовою програмування C++ **та вміння:** визначати необхідність та програмувати дружні класи, методи та функції, шаблони функцій та шаблони класів; перевантажувати шаблони та оператори; обробляти помилки та виняткові ситуації; програмувати побудову примітивних 2-Д та 3-Д зображень; задавати дизайн Windows-додатків; програмувати Windows-додатки з використанням певних компонент мовою програмування C# в середовищі розробки Visual Studio; використовувати розглянутий інструментарій візуального (кросплатформного) середовища Qt для розробки графічного інтерфейсу додатків для задач різного спрямування.

### 4. Результати навчання (Компетентності) відповідно до освітньо-професійної програми 2021 р. спеціальності 122:

1. **ЗК1.** Здатність до абстрактного мислення, аналізу та синтезу.
2. **ЗК2.** Здатність застосовувати знання у практичних ситуаціях.
3. **ЗК3.** Знання та розуміння предметної області та розуміння професійної діяльності.
4. **ЗК6.** Здатність вчитися й оволодівати сучасними знаннями.
5. **ЗК8.** Здатність генерувати нові ідеї (креативність).
6. **ЗК9.** Здатність працювати в команді.
7. **СК3.** Здатність до логічного мислення, побудови логічних висновків, використання формальних мов і моделей алгоритмічних обчислень, проектування, розроблення й аналізу алгоритмів, оцінювання їх ефективності та складності, розв’язності та нерозв’язності алгоритмічних проблем для адекватного моделювання предметних областей і створення програмних та інформаційних систем.
8. **СК8.** 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.
9. **ПРН5.** Проектувати, розробляти та аналізувати алгоритми розв’язання обчислювальних та логічних задач, оцінювати ефективність та складність алгоритмів на основі застосування формальних моделей алгоритмів та обчислювальних функцій.
10. **ПРН9.** Розробляти програмні моделі предметних середовищ, вибирати парадигму програмування з позицій зручності та якості застосування для реалізації методів та алгоритмів розв’язання задач в галузі комп’ютерних наук.

## 5. Структура навчальної дисципліни

### I семестр

Назви змістових модулів і тем	Усього	Лек.	Лабор.	Сам. роб.	Конс.	Форма контролю/ Бали
<b>Змістовий модуль 1. Продовження ООП: дружні класи, шаблони класу, запис об'єкта у файл. Використання графічних бібліотек</b>						
Тема 1. Дружні функції, методи та класи у C++.		2	2	3		IPC/1,5
Тема 2. Шаблони функцій та класів у C++.		2	2	3		IPC/1,5
Тема 3. Шаблон функції з різними варіантами змінних та перевантаження шаблонів у C++.		2	4	3		IPC/3
Тема 4. Шаблони: перевантаження операторів.		2	2	3		IPC/1,5
Тема 5. Файли (повторення) у C++. Довільне записування та зчитування з файлу. Бінарні файли. Запис об'єкта у файл та читання з файлу.		2	2	4	1	IPC/2
Тема 6. Графіка у C++ (початок): налаштування, основні функції та принципи.		2	2	4		IPC/1,5
Тема 7. Приклади побудови графічних зображень, рух простого зображення. Графік функції, рух складного зображення.		2	6	4	1	IPC/5
Тема 8. Графічна бібліотека OpenGL.		2	2	4		IPC/1,5
Тема 9. Побудова, зафарбовування та обертання примітивів з використанням OpenGL		2	2	4	1	IPC/2
Тема 10. Створення 3D-об'єктів вручну, використовуючи вбудовані функції та з використанням масивів		2	4	3	1	IPC/1,5
<b>Разом за модулем 1</b>	<b>87</b>	<b>20</b>	<b>28</b>	<b>35</b>	<b>4</b>	<b>IPC/21</b>
<b>Змістовий модуль 2. Середовище Visual Studio: створення додатків з використанням компонентів</b>						
Тема 11. Основні принципи візуального програмування.		2	2	4		IPC/1

Встановлення середовища Visual Studio Community. Форма, обробник події, компоненти.						
Тема 12. Робота у Visual Studio Community. Компонент MessageBox та Form. Створення найпростіших Windows-додатків в середовищі Visual Studio з використанням С#		2	2	4		IPC/1,5
Тема 13. Компонент кнопка Button, мітка Label, посилання LinkLabel, текстове поле TextBox. Робота з компонентами. Розгалуження та цикли у С#		2	4	5	1	IPC/3,5
Тема 14. Компоненти Radiobutton, CheckBox, ListBox, ComboBox, PictureBox для програмування інтерфейсу у середовищі Visual Studio Community		2	4	5	1	IPC/3
Тема 15. Компоненти MaskedTextBox, NumericUpDown, DomainUpDown, ListView для програмування інтерфейсу у середовищі Visual Studio Community		2	4	4	1	IPC/3,5
Тема 16. Компоненти TreeView, DataGridView їх властивості та використання		2	4	4	1	IPC/3,5
Тема 17. Компонент Chart, його властивості, методи та використання		2	2	4		IPC/1,5
Тема 18. Програмування подій MouseDown, MouseUp, MouseMove. (Програмне формування компонента Label на формі використовуючи дані події)		2	4	4		IPC/1,5
<b>Разом за модулем 2</b>	<b>80</b>	<b>16</b>	<b>26</b>	<b>34</b>	<b>4</b>	<b>IPC/19</b>
<b>Види підсумкових робіт</b>						Бал
Контрольна робота № 1 (тести)						15
Контрольна робота № 2 (тести)						15
Індивідуальна робота студента (IPC)						20
За відвідування лекцій						10
<b>Всього годин/ Балів</b>	<b>167</b>	<b>36</b>	<b>54</b>	<b>69</b>	<b>8</b>	<b>100</b>

II семестр

Назви змістових модулів і тем	Усього	Лек.	Лабор.	Сам. роб.	Конс.	Форма контролю/ Бали
<b>Змістовий модуль 1. Середовище програмування Qt: основні можливості та інструменти, класи віджетів</b>						
Тема 1. Знайомство з середовищем програмування Qt. Інтегроване середовище розробки Qt Creator. Ієрархія класів віджетів. Консольний та графічний проекти		2	2	7		IPC/2,5
Тема 2. Створення графічного інтерфейсу засобами Qt: менеджери компоновки (layout managers). Віджети, їх характеристики, налаштування стилю		2	4	7		IPC/4
Тема 3. Qt Designer – швидка розробка прототипів. Слоти та сигнали		2	2	6	1	IPC/2,5
Тема 4. Клас QMenu – меню та його види. Використання віджетів timeEdit, dateEdit, dateTimeEdit, calendarWidget для створення користувацького інтерфейсу		2	4	7	1	IPC/5
Тема 5. Діалогові вікна для створення користувацького інтерфейсу. Клас QDialog		2	2	7	1	IPC/2
Тема 6. Інструменти для створення інтерфейсу та функціоналу текстового редактору. Створення користувацького віджету		2	2	6		IPC/4,5
Тема 7. Модульне тестування коду на C++ за допомогою Qt Test. Вступ в тестування, розширене тестування		2	4	7	1	IPC/4,5
Тема 8. Модульне тестування GUI (графічного інтерфейсу користувача) за допомогою Qt Test. Вступ в тестування, розширене тестування		2	4	7		IPC/4,5
<b>Разом за модулем 1</b>	<b>98</b>	<b>16</b>	<b>24</b>	<b>54</b>	<b>4</b>	<b>IPC/27,5</b>

<b>Змістовий модуль 2. Патерни проектування</b>						
Тема 9. Патерни проектування: породжуючі патерни		2	2	5		IPC/2,5
Тема 10. Патерни проектування: структурні патерни		2	2	5	1	IPC/2,5
Тема 11-12. Патерни проектування: поведінкові патерни		4	6	5	1	IPC/7,5
<b>Разом за модулем 2</b>	<b>35</b>	<b>8</b>	<b>10</b>	<b>15</b>	<b>2</b>	<b>IPC/12,5</b>
<b>Види підсумкових робіт</b>						Бал
Модульна контрольна робота № 1						20
Модульна контрольна робота № 2						20
Індивідуальна робота студента (IPC)						20
<b>Всього годин/ Балів</b>	<b>133</b>	<b>24</b>	<b>34</b>	<b>69</b>	<b>6</b>	<b>100</b>

## 6. Завдання для самостійного опрацювання

I семестр

<b>№</b>	<b>Тема</b>
1.	Обробка помилок на різних етапах розв'язування та програмування завдань
2.	Шаблони функцій: на вхід подається 2 параметри узагальненого типу
3.	Шаблони функцій: на вхід подається 2 параметри узагальненого типу, один з яких клас
4.	Шаблони функцій з локальною змінною узагальненого типу
5.	Перевантаження шаблонів
6.	Програмування довільного запису різної інформації у файли
7.	Запис, дозапис та читання різних об'єктів класу у файли
8.	Середовище Visual Studio та особливості роботи при створенні додатків
9.	Програмування лінійних, розгалужених та циклічних програм мовою C#
10.	Компоненти та їх властивості (режим app) у середовищі Visual Studio, обробники подій та їх особливості.
11.	Дизайн форм для додатків у середовищі Visual Studio (режим app).
12.	Виконання завдань до лабораторних робіт
13.	Підготовка лекційного матеріалу, перегляд додаткового відео та Інтернет-джерел
14.	Опрацювання матеріалу для підготовки до заліку

II семестр

<b>№</b>	<b>Тема</b>
1.	Підготовка лекційного матеріалу, перегляд додаткового відео та Інтернет-джерел
2.	Виконання завдань до лабораторних робіт
3.	Опрацювання матеріалу для підготовки до екзамену



## **IV. Політика оцінювання**

### **Політика викладача щодо студента**

Усі учасники освітнього процесу повинні дотримуватись вимог чинного законодавства України, Статуту і Правил внутрішнього розпорядку ВНУ імені Лесі Українки, загально-прийнятих моральних принципів, правил поведінки та корпоративної культури; підтримувати атмосферу доброзичливості, відповідальності, порядності й толерантності. Атмосфера на заняттях повинна бути творчою, відкритою до конструктивної критики. Недопустимі запізнення на заняття; користування мобільним телефоном, планшетом чи іншими мобільними пристроями під час заняття; списування. Очікується, що всі студенти відвідають усі лекції і практичні заняття курсу.

### **Політика щодо академічної доброчесності**

Дотримання академічної доброчесності здобувачами передбачає: самостійне виконання завдань поточного та підсумкового контролю (для осіб з особливим освітніми потребами ця вимога застосовується з урахуванням їх індивідуальних потреб і можливостей); посилання на джерела інформації у разі використання ідей, тверджень, відомостей; дотримання норм законодавства про авторське право; надання достовірної інформації про результати власної навчальної (наукової, творчої) діяльності.

Порушенням академічної доброчесності вважається: академічний плагіат, самоплагіат, фабрикація, фальсифікація, списування. За порушення академічної доброчесності здобувачі освіти можуть бути притягнені до такої академічної відповідальності: повторне проходження оцінювання; повторне проходження відповідного освітнього компонента освітньої програми.

Під час модульного та підсумкового контролю (заліку) студентам заборонено користуватись такими засобами як мобільний телефон, планшет, конспект, навчальна література, інші джерела інформації, в тому числі Інтернет-ресурси.

### **Політика щодо дедлайнів та перескладання**

Усі передбачені завдання мають бути виконані у встановлений термін. Несвоєчасно виконані завдання оцінюються на нижчу оцінку. Виключенням можуть бути завдання, які не вдалося зробити з поважних причин, в такому випадку студент може доробити вказані завдання у вказаний термін.

Якщо здобувач вищої освіти був відсутній на заняттях з будь-якої причини, то він (вона) вивчає матеріал самостійно, використовуючи навчальні посібники, конспекти лекцій, матеріали дистанційного курсу, у випадку розміщення його на платформі дистанційного навчання Moodle, виконує всі домашні завдання. Прозвітуватися про виконання завдань можна, використовуючи дистанційний курс, прикріпивши виконанні завдання у відповідні комірки та попередити викладача про здане завдання, або під час консультацій або надіслати виконане завдання на корпоративну пошту викладача. Зворотній зв'язок з викладачем для з'ясування всіх питань: використання форуму, чату дистанційного курсу, корпоративної пошти університету або відповідної бесіди у певному месенджері.

Перескладання модульного контролю (письмового чи тестування) заборонено.

## V. Підсумковий контроль

### I семестр

Підсумковий контроль з даної дисципліни передбачено у вигляді заліку.

Оцінювання здійснюється за 100-бальною шкалою. Оцінка включає в себе поточний контроль, який нараховується за виконання лабораторних робіт, індивідуальних робіт, контрольних робіт, тестових робіт до лекційних матеріалів курсу. Максимальна кількість балів, яку може отримати студент під час поточного оцінювання за семестр – 100 балів. Якщо за результатами семестру студент отримав не менше 60 балів і погоджується із цим результатом, то оцінка за семестр виставляється без складання заліку. В іншому випадку студент складає залік на ліквідації; максимальна кількість балів, яку можна отримати – 100 балів.

Порядок проведення заліку-ліквідації – залік відбувається у вигляді виконання практичного завдання та відповіді на теоретичні питання.

### II семестр

Підсумковий контроль з даної дисципліни передбачено у вигляді екзамену.

Оцінювання здійснюється за 100-бальною шкалою. Оцінка включає в себе поточний контроль та підсумковий модульний контроль. Максимальна кількість балів, яку може отримати студент під час поточного оцінювання за семестр – 40 балів. Підсумковий модульний контроль за семестр включає в себе оцінки за всі модульні контрольні роботи, тестові завдання, індивідуальну роботу студента і складає 60 балів. Якщо студент за період вивчення дисципліни набрав за поточний та модульний контроль мінімум 75 балів і погоджується із цим результатом, то оцінка за семестр може виставлятися без складання екзамену. В іншому випадку студент складає екзамен; максимальна кількість балів, яку можна отримати на екзамені – 60 балів, але бали за підсумковий модульний контроль анулюються. Оцінка за семестр, у випадку складання екзамену, є сумою балів поточного контролю та балів, отриманих під час екзамену.

Порядок проведення екзамену – екзамен відбувається у вигляді виконання практичного завдання та відповіді на теоретичні питання.

## VI. Шкала оцінювання

**Шкала оцінювання знань здобувачів освіти, де формою контролю є іспит**

Оцінка в балах	Лінгвістична оцінка	Оцінка за шкалою ECTS	
		оцінка	Пояснення
90–100	Відмінно	A	відмінне виконання
82–89	Дуже добре	B	вище середнього рівня
75–81	Добре	C	загалом хороша робота
67–74	Задовільно	D	Непогано
60–66	Достатньо	E	виконання відповідає мінімальним критеріям
1–59	Незадовільно	Fx	Необхідне перескладання

### Шкала оцінювання знань здобувачів освіти, де формою контролю є залік

Оцінка в балах	Лінгвістична оцінка
90–100	Зараховано
82–89	
75–81	
67–74	
60–66	
1–59	Незараховано (необхідне перескладання)

### VI. Рекомендована література та Інтернет-ресурси I семестр

1. Програмування: підручник [Електронний ресурс] / укладач Л. Я. Глинчук, Т.О. Гришанович; ВНУ ім. Лесі Українки. – Електронні текстові данні (1 файл: 3 201 КБ). – Луцьк: ВНУ ім. Лесі Українки, 2021. – 160 с.
2. Власюк А.П., Мартинюк П.М., Прищепя О.В., Філатова І.А., Філатов М.С., Рощенюк А.М., Демчук О.С., Демчук М.Б., Мічута О.Р., Цветкова Т.П., Федорчук Н.А. Лабораторний практикум з програмування. Навч. посібник / За загальною редакцією проф. Власюка А.П. – Рівне: НУВГП, 2011. – 495 с.
3. Васильев А.Н., Самоучитель С++ с примерами и задачами. Книга + CD. – СПб.: Наука и техника, 2010. – 480 с.: ил. (+CD)
4. Васильев А. Н. Программирование на С++ в примерах и задачах/ Алексей Васильев. – Москва: Издательство “Э”, 2017. – 368 с.
5. Блинов, И. Н. Объектно-ориентированное программирование на языке С++: учебно-метод. пособие для студ. механико-матем. фак. / И. Н. Блинов, В. С. Романчик. Мн.: БГУ, 2005. – 57 с.
6. Албахари Джозеф, Албахари Бен. С# 7.0. Карманный справочник.: Пер. с англ. – СПб.: ООО «Альфа-книга», 2017. – 224 с.
7. Мюллер, Джон Пол, Семф, Билл, Сфер, Чак. С# для чайников.: Пер. с англ. - СПб.: ООО "Диалектика", 2019. - 608 с.: ил.
8. Стилмен Э., Грин Дж. Изучаем С#. 3-е изд. – СПб.: Питер, 2014. – 816 с.: ил. – (Серия «Head First O’Reilly»). ISBN 978-5-496-00867-9
9. Загребельный С.Л., Основы визуального программирования в среде Visual Studio 2010: учебное пособие для студентов / С.Л. Загребельный, А.А. Костиков, В.Э. Миринский. – Краматорск: ДГМА, 2012. – 160 с. ISBN 978-966-379-599-7.

### Додаткова література (Інтернет-джерела)

1. С# програмування: лекції та практичні. [Електронний ресурс]. – Режим доступу до ресурсу: <http://www.e-helper.com.ua/node/941/?categories=term/39>
2. Встановлення версії Visual Studio Community (безкоштовна версія), опис середовища та його вікон. [Електронний ресурс]. – Режим доступу до ресурсу:

[https://www.youtube.com/watch?v=gp2rA0rgq\\_0&ab\\_channel=%D0%93%D0%BE%D1%88%D0%B0%D0%94%D1%83%D0%B4%D0%B0%D1%80%D1%8C](https://www.youtube.com/watch?v=gp2rA0rgq_0&ab_channel=%D0%93%D0%BE%D1%88%D0%B0%D0%94%D1%83%D0%B4%D0%B0%D1%80%D1%8C)

3. Створення найпростішого Windows-додатка на C# (сума двох чисел). [Електронний ресурс]. – Режим доступу до ресурсу:  
[https://www.youtube.com/watch?v=LiYM9VW7mCo&ab\\_channel=%D0%98%D0%B7%D1%83%D1%87%D0%B0%D0%B5%D0%BC%D0%BC%D0%B8%D1%80%D0%98%D0%A2%2F%D0%9E%D0%BB%D0%B5%D0%B3%D0%A8%D0%BF%D0%B0%D0%B3%D0%B8%D0%BD%2FWISEPLAT](https://www.youtube.com/watch?v=LiYM9VW7mCo&ab_channel=%D0%98%D0%B7%D1%83%D1%87%D0%B0%D0%B5%D0%BC%D0%BC%D0%B8%D1%80%D0%98%D0%A2%2F%D0%9E%D0%BB%D0%B5%D0%B3%D0%A8%D0%BF%D0%B0%D0%B3%D0%B8%D0%BD%2FWISEPLAT)
4. Створення дизайну для вікна Windows-додатка на C#. [Електронний ресурс]. – Режим доступу до ресурсу:  
[https://www.youtube.com/watch?v=gnvxVv60xnI&ab\\_channel=%D0%93%D0%BE%D1%88%D0%B0%D0%94%D1%83%D0%B4%D0%B0%D1%80%D1%8C](https://www.youtube.com/watch?v=gnvxVv60xnI&ab_channel=%D0%93%D0%BE%D1%88%D0%B0%D0%94%D1%83%D0%B4%D0%B0%D1%80%D1%8C)

## II семестр

1. Шлее М. Qt 4.8. Профессиональное программирование на C++. – СПб.: БХВ-Петербург, 2012. – 912 с.: ил. – (В подлиннике).
2. Шлее М. Qt 4.8. Профессиональное программирование на C++. – СПб.: БХВ-Петербург, 2018. – 1074 с.: ил. – (В подлиннике).
3. Программирование на языке C++ в среде Qt Creator: / Е.Р. Алексеев, Г.Г. Злобин, Д.А. Костюк, О.В. Чеснокова, А.С. Чмыхало – М.: ALT Linux, 2015. – 448с.: ил.
4. Будай А. Дизайн-патерни — просто, як двері. 2012. – 90 с. (електронна книга)

### Додаткова література (Інтернет-джерела)

1. Програмування у Qt мовою C++. [Електронний ресурс]. – Режим доступу до ресурсу: [https://www.youtube.com/watch?v=DOGc4PxklI0&list=PLW1\\_G4jip1oPmiH-G8EBczKRcNKrij970&ab\\_channel=%D0%92%D0%BE%D0%BB%D0%BE%D0%B4%D0%B8%D0%BC%D0%B8%D1%80%D0%91%D0%BE%D0%BD%D0%B4%D0%B0%D1%80%D0%B5%D0%BD%D0%BA%D0%BE](https://www.youtube.com/watch?v=DOGc4PxklI0&list=PLW1_G4jip1oPmiH-G8EBczKRcNKrij970&ab_channel=%D0%92%D0%BE%D0%BB%D0%BE%D0%B4%D0%B8%D0%BC%D0%B8%D1%80%D0%91%D0%BE%D0%BD%D0%B4%D0%B0%D1%80%D0%B5%D0%BD%D0%BA%D0%BE)
2. Уроки Qt Creator. [Електронний ресурс]. – Режим доступу до ресурсу: <https://itproger.com/course/qt-creator>
3. Цикл уроків по Qt. [Електронний ресурс]. – Режим доступу до ресурсу: <https://www.youtube.com/playlist?list=PLbmlzoDQrXVHDN2nBogk7qy-4DzgyNhNq>
4. Уроки по Qt. [Електронний ресурс]. – Режим доступу до ресурсу: <https://ravesli.com/uroki-po-qt5/>
5. 265 кроків для освоєння Qt. [Електронний ресурс]. – Режим доступу до ресурсу: <http://it.kgsu.ru/Qt/oglav.html>
6. Патерни проектування. [Електронний ресурс]. – Режим доступу до ресурсу: <https://refactoring.guru/uk/design-patterns>

## Питання та завдання до заліку-ліквідації I семестр

### Теоретична частина

1. До яких елементів класу доступ існує тільки всередині класу?
2. Які функції називаються дружніми?
3. Як оголосити функцію дружню до класу?
4. Що таке прототип функції?
5. Яку можливість в описі класу дає даний рядок «friend void show (MyClass obj)»?
6. Опишіть випадок коли доцільним є застосування дружніх функцій?
7. В якому полі класу може бути розміщена дружня функція?
8. Що таке анонс класу та як він оголошується?

9. У яких випадках потрібно оголошувати анонс класу?
10. Чи можуть бути окремі методи класу дружніми по відношенню до іншого класу?
11. Наведіть приклад, де всі методи дружнього класу мають доступ до закритих полів та методів іншого класу.
12. Як оголосити дружній метод? Наведіть приклад коду.
13. Якщо у класі B є інструкція friend class A, то що вона означає?
14. У якому випадку всі методи класу A мають доступ до елементів класу B, в тому числі і закритих?
15. Коли до функції можна буде звернутися в не класу, не використовуючи спеціальних методів?
16. Чи може дружня функція отримувати вказівник this? Обґрунтуйте відповідь.
17. Як викликаються дружні функції?
18. В якому полі класу можна оголосити інший клас як дружній?
19. Чи можемо ми звертатися до методів дружнього класу з головної функції main ()?
20. Коли клас оголошено дружнім, всі його методи також ...до того класу в якому він оголошений. Приклад.
21. Що таке шаблон функції?
22. Наведіть приклад, коли доцільно використати шаблон функції.
23. Яке службове слово показує, що оголошено шаблон функції?
24. Що записують після службового слова в шаблоні функції?
25. Якщо використовувати шаблони, то код програми збільшиться чи зменшиться? Чому?
26. Наведіть приклад використання шаблону функції, відмінний від прикладу в лекції.
27. Як впливає на пам'ять використання шаблону функції?
28. Який загальний вигляд має шаблон класу?
29. Коли доцільно використовувати шаблон класу?
30. Наведіть приклад шаблону класу, відмінний від прикладу в лекції.
31. Наведіть приклад шаблону з одним аргументом.
32. Поясніть суть шаблону з одним аргументом.
33. Наведіть приклад шаблону (узагальненої функції) з локальними змінними узагальненого типу.
34. Поясніть суть шаблону з локальними змінними узагальненого типу.
35. Яке ключове слово вказується перед кожним узагальненим типом у шаблоні функції?
36. Наведіть приклади використання узагальненого (шаблону) класу з узагальненим типом клас.
37. Опишіть особливості узагальненого класу з використанням узагальненого (шаблону) типу клас.
38. Прокоментуйте приклад, де узагальнений клас використовує 2 узагальнені типи, один з яких клас, вкажіть особливості.
39. Що таке перевантаження функцій?
40. Чи можна виконувати перевантаження для шаблонів функцій? Обґрунтуйте.
41. Наведіть приклади використання перевантаження шаблонів функцій.
42. Як відбувається перевантаження оператора? Наведіть приклад використання перевантаження при узагальненому класі.
43. Для чого служить клас ifstream, ofstream, fstream?
44. Як створюються об'єкти цих класів для записування (відкриття) вихідного потоку?
45. Як створюються об'єкти цих класів для зчитування вихідного потоку?
46. Як створюються об'єкти цих класів для введення-виведення (записування і зчитування) вихідного потоку?
47. Які операції використовують для того аби «помістити у потік», «взяти з потоку»?
48. Які ви знаєте режими відкривання файлів? Що означає кожен з режимів?

49. Наведіть приклади коду відкривання файлів з різними режимами.
50. Що таке довільне записування у файл та зчитування даних з файлу та як воно відбувається?
51. За допомогою якого методу відбувається позиціонування курсора у файлі? Наведіть та прокоментуйте приклади коду позиціонування курсора у файлі.
52. Які методи повертають поточну позицію файлу для потоків введення та виведення?
53. Якої довжини повинні бути рядки, що записані у файл? Що це дає? Наведіть приклади.
54. Як записати у файл об'єкт будь-якого користувацького класу? Як зчитати з файлу об'єкт будь-якого користувацького класу?
55. Які файли називають бінарними? Яка відмінність бінарного файлу від текстового?
56. Який прапорець показує, що робота буде вестися з бінарним форматом файлу?
57. За допомогою якого методу дані записуються/зчитуються у бінарний файл?
58. Як у бінарному файлі організована можливість довільного доступу?
59. Чи буде працювати компілятор з графікою без допоміжних налаштувань?
60. Як на графічному екрані утворюється зображення? Як називаються точки на графічному екрані? Від чого залежить кількість точок на екрані?
61. З якої частини екрану ведуть відлік точок? Чим характеризується кожна точка?
62. З чого складаються програми, що виконують графічні побудови?
63. Опишіть як перейти працювати у графічний режим.
64. Для чого призначена функція `initgraph(&gdriver, &gmode, "")` та що задає? Опишіть кожен аргумент та його можливе значення з функції `initgraph(&gdriver, &gmode, "")`?
65. Для чого використовується функція `graphresult()`? Для чого використовується функція `getch()`? Для чого використовується функція `closegraph()`?
66. Назвіть функції для роботи з кольором.
67. Опишіть алгоритм (інструкцію) налаштування графічного режиму.
68. Назвіть основні функції для побудови точки. Назвіть основні функції для побудови лінії.
69. Назвіть основні функції для побудови різних фігур.
70. Як задати заповнення та стиль для фігур? Для чого призначена функція `floodfill()`?
71. Які функції у графічному режимі призначені для роботи з текстом?
72. Які функції дозволяють отримати координати центру екрану?
73. Наведіть приклад побудови еліпса та заповнення його певним кольором та певним стилем.
74. Як розмістити текст у графічному режимі горизонтально, вертикально?
75. Які стилі ви знаєте для виведення тексту, ліній?
76. Опишіть ідею створення рухомого зображення?
77. Наведіть приклад створення руху елементарної фігури.
78. На які частини можна розбити програмування побудови графіка функції?
79. Як побудувати осі координат? Які функції потрібно використати для побудови координатних осей? Як підписати координатні осі?
80. Що таке масштабування та для чого його використовують?
81. Як запрограмувати масштабування графіка функції?
82. Коли використовуються цикли у графіці?
83. Чому ми розрізняємо просте графічне зображення та складне? Наведіть приклади простого та складного.
84. У яких випадках ми говоримо про рух простої/складної фігури?
85. В чому полягає іде створення руху складного зображення?
86. Що таке OpenGL? Охарактеризуйте стандарт OpenGL.
87. Які ОС підтримують використання OpenGL?
88. Опишіть характеристики OpenGL.

89. У вигляді чого в OpenGL представляються об'єкти?
90. Побудову який графічних примітивів забезпечують базові функції?
91. Для формування яких зображень призначені функції з розширення базового набору?
92. Які додаткові бібліотеки містить OpenGL?
93. Для чого призначені бібліотеки утиліт GL, GLU?
94. Для чого призначені бібліотеки утиліт GLUT, GLX?
95. Опишіть архітектуру OpenGL.
96. Прокоментуйте схему функціонування конвеєра OpenGL.
97. Які особливості OpenGL пов'язані з апаратним рівнем та рівнем користувача?
98. OpenGL можна розглядати як скінченний автомат. Чому? Обґрунтуйте.
99. Як почати використовувати бібліотеки OpenGL у C++?
100. Яка особливість команд бібліотеки OpenGL? Що таке префікс та суфікс у командах OpenGL? Який вигляд має повне ім'я команди в OpenGL?
101. Опишіть алгоритм налаштувань графічної бібліотеки OpenGL.
102. Прокоментуйте початковий код з ініціалізації бібліотеки.
103. Розгляньте приклад побудови трикутника, як задають вершини та кольори?
104. Яка технологія програмування побудови фігури використовується і у цій графічній бібліотеці?
105. Для чого служить функція `glutInit()`? Для чого служить функція `glutInitDisplayMode()`?
106. Для чого служить функція `glutInitWindowSize()`? Для чого служить функція `glutCreateWindow()`?
107. Для чого служать функції: `glutDisplayFunc(Display)`; `glutReshapeFunc(Reshape)`; `glutKeyboardFunc(Keyboard)`? Яка функція контролює усі події та виклики інших функцій?
108. Що таке статичне зображення, динамічне зображення?
109. Що забезпечує типова функція оновлення зображення?
110. Для чого програма викликає команду `glClear`?
111. Що таке вершина і чим вона характеризується в OpenGL? Прокоментуйте команду `glVertex`?
112. Які атрибути та команди пов'язані з кольором та текстурою?
113. Які атрибути та команди для точок, ліній. Чотирикутників, багатокутників?
114. Для чого використовуються так звані операторні дужки?
115. Як відбувається визначення примітива чи послідовності примітивів?
116. Які примітиви дозволяє будувати OpenGL?
117. У яких режимах можна побудувати трикутник, прямокутник?
118. Які системи координат використовуються в OpenGL? Яка система координат вважається основною?
119. Як саме формується спосіб представлення системи координат за правилом правої руки?
120. Як вибрати координати, щоб трикутник побудувати по центру екрану?
121. Яку функцію потрібно викликати, щоб встановити початок системи координат?
122. Яка функція в OpenGL використовується для переміщення примітивів?
123. Як відбувається переміщення осей координат?
124. Як задається колір для примітиву, які ви знаєте види фарбувань?
125. Розгляньте та прокоментуйте приклад фарбування чотирикутників градієнтом.
126. За допомогою якої функції виконується обертання примітивів вздовж осі та переміщення?
127. Як відбувається побудова, зміщення та поворот фігури, коли всі точки будуються вручну?
128. Які особливості повороту, зміщення та повороту куба?

129. Яка функція дозволяє масштабування? Прокоментуйте вхідні аргументи функції масштабування? Як вона працює?
130. Яка функція будує каркас кубу? Яка функція будує зафарбований куб?
131. Які команди потрібно використати аби не порушити систему координат?
132. Які інші 3D-об'єкти можна будувати?

### **Приклади практичних завдань**

1. Продемонструйте як використовують компонент MessageBox, його властивості.
2. Продемонструйте застосування/зміни основних властивостей компонента Form.
3. Продемонструйте програмування форми не прямокутного вигляду.
4. Покажіть роботу з 2-ма формами.
5. Продемонструйте застосування/зміни основних властивостей компонента Button. Пропишіть код для компонента Button та продемонструйте його роботу.
6. Особливості компонента Label та посилання LinkLabel. Основні властивості та приклади використання.
7. Особливості компонента TextBox. Основні властивості та приклад використання.
8. Особливості компонента Radiobutton. Основні властивості та приклад використання.
9. Особливості компонента CheckBox. Основні властивості та приклад використання.
10. Особливості компонента ListBox. Основні властивості та приклад використання.
11. Особливості компонента ComboBox. Основні властивості та приклад використання.
12. Особливості компонента PictureBox. Основні властивості та приклад використання.
13. Особливості компонента MaskedTextBox. Основні властивості та приклад використання.
14. Особливості компонента NumericUpDown. Основні властивості та приклад використання.
15. Особливості компонента DomainUpDown. Основні властивості та приклад використання.
16. Особливості компонента ListView. Основні властивості та приклад використання.
17. Особливості компонента TreeView. Основні властивості та приклад використання.
18. Особливості компонента DataGridView. Основні властивості.
19. Приклад використання компонента DataGridView.
20. Особливості компонента Chart. Основні властивості та приклад використання для побудови графіка функції.
21. Особливості компонента Chart. Основні властивості та приклад використання для побудови діаграм.
22. Програмування подій MouseDown, MouseUp. Приклад.
23. Програмування події MouseMove. Приклад.

### **Питання до екзамену II семестр**

1. Основні складові Qt. Приклади ПЗ, створеного з використанням Qt.
2. Ліцензування Qt, довідка та ресурси. Огляд налаштувань середовища Qt.
3. Перші проекти: консольний додаток Qt та додаток Qt Widgets
4. Детальна структура Qt-проекту.
5. Користувацький інтерфейс та всі засоби Qt Creator. Інтерактивний відладчик, типи помилок.
6. Менеджери компоновки (layout managers), клас QLayout.
7. Клас QBoxLayout та його основні методи.
8. Горизонтальне розміщення QHBoxLayout.
9. Вертикальне розміщення QVBoxLayout.



10. Вкладені розміщення.
11. Табличне розміщення QGridLayout.
12. Віджети, їх характеристики, налаштування стилю.
13. Механізм сигналів та слотів.
14. Метод connect () класу QObject.
15. Автоматичне поєднання слота і сигналу за допомогою режиму «Редагувати сигнали/слоти».
16. Створення (поєднання) слота і сигналу за допомогою «перейти до слота».
17. Клас QMenu – меню та його види.
18. Об'єкти класів: QDate, QTime и QDateTime.
19. Об'єкт класу QTimer – таймер.
20. Клас QDateTimeEdit.
21. Віджети timeEdit, dateEdit, dateTimeEdit, calendarWidget.
22. Правила створення діалогових вікон.
23. Клас QDialog. Модальні, немодальні діалогові вікна.
24. Створення власного діалогового вікна. Екранні заставки.
25. Діалогові вікна, що міняються.
26. Стандартні діалогові вікна. Діалогові вікна повідомлень.
27. Особливості віджету однорядкове текстове поле класу QLineEdit.
28. Редактор тексту, клас QTextEdit. (+ Клас QTextCursor).
29. Запис у файл. (Клас QPdfWriter)
30. Файл ресурсів, картинка біля дій в головному меню, панель інструментів toolBar – швидкий доступ до інструментів.
31. Буфер обміну - клас QClipboard.
32. Відтворення WAV-файлів: клас QSound.
33. Більше можливостей відтворення звукових файлів – клас QMediaPlayer.
34. Створення користувацького віджету.
35. Що таке патерн? Історія патернів. Навіщо знати патерни? Класифікація патернів.
36. Породжуючий патерн: Абстрактна Фабрика (Abstract Factory).
37. Породжуючий патерн: Будівельник (Builder).
38. Породжуючий патерн: Фабричний метод (Factory Method).
39. Породжуючий патерн: Прототип (Prototype).
40. Породжуючий патерн: Одинак (Singleton).
41. Структурний патерн: Адаптер (Wrapper, Обгортка, Adapter).
42. Структурний патерн: Міст (Bridge).
43. Структурний патерн: Компонувальник (Дерево, Composite).
44. Структурний патерн: Декоратор (Wrapper, Обгортка, Decorator).
45. Структурний патерн: Фасад (Facade).
46. Структурний патерн: Легковаговик (Пристосованець, Кеш, Flyweight).
47. Поведінковий патерн: Ланцюжок обов'язків (Ланцюг відповідальностей, CoR, Chain of Command, Chain of Responsibility).
48. Поведінковий патерн: Команда (Дія, Транзакція, Action, Command).
49. Поведінковий патерн: Ітератор (Iterator).
50. Поведінковий патерн: Посередник (Intermediary, Controller, Mediator).
51. Поведінковий патерн: Знімок (Memento).
52. Поведінковий патерн: Структурний патерн: замісник (Proxy).
53. Поведінковий патерн: Спостерігач (Observer).
54. Поведінковий патерн: Стан (State).

55. Поведінковий патерн: Стратегія (Strategy).
56. Поведінковий патерн: Шаблонний метод (Template Method).
57. Поведінковий патерн: Відвідувач (Visitor).

#### **Приклади практичних завдань**

1. Продемонструвати створення проектів: консольний додаток Qt та додаток Qt Widgets та детальна структура Qt-проекту.
2. Продемонструвати роботу з менеджером компоновки (layout managers), клас QLayout (горизонтальне, вертикальне розміщення).
3. Продемонструвати роботу з менеджером компоновки (layout managers), клас QLayout (вкладене, табличне розміщення).
4. Продемонструвати автоматичне поєднання слота і сигналу за допомогою режиму «Редагувати сигнали/слоти».
5. Продемонструвати створення (поєднання) слота і сигналу за допомогою «перейти до слота».
6. Продемонструвати роботу з класом QMenu.
7. Продемонструвати використання об'єктів класів: QDate, QTime и QDateTime.
8. Продемонструвати роботу з віджетами timeEdit, dateEdit, dateTimeEdit, calendarWidget. Їх властивості та використання.
9. Продемонструвати побудову коду на мові програмування за породжуючим патерном: Фабричний метод (Factory Method).
10. Продемонструвати побудову коду на мові програмування за структурним патерном: Фасад (Facade).
11. Продемонструвати побудову коду на мові програмування за поведінковим патерном: Стратегія (Strategy).