

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ  
ЗАПОРІЗЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ  
МАТЕМАТИЧНИЙ ФАКУЛЬТЕТ  
КАФЕДРА ПРОГРАМНОЇ ІНЖЕНЕРІЇ

**ЗАТВЕРДЖУЮ**

Декан математичного факультету

\_\_\_\_\_ Гоменюк С. І.  
(підпис) (прізвище, ініціали)

“ \_\_\_\_ ” \_\_\_\_\_ 2019 р.

**ТЕХНОЛОГІЇ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ**

**РОБОЧА ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ**

підготовки бакалаврів  
спеціальності 113 – “Прикладна математика”  
освітньо-професійної програми “Прикладна математика”

**Укладач:** д.т.н., професор, професор кафедри програмної інженерії Гоменюк С. І.

Обговорено та ухвалено  
на засіданні кафедри програмної інженерії

Ухвалено науково-методичною радою  
математичного факультету

Протокол № 1 від “20” серпня 2019 р.  
Завідувач кафедри \_\_\_\_\_ Лісняк А. О.

Протокол № 1 від “30” серпня 2019 р.  
Голова науково-методичної  
ради математичного  
факультету \_\_\_\_\_ Пшенична О. С.

2019 рік

## 1. Опис навчальної дисципліни

Найменування показників	Галузь знань, напрям підготовки/ спеціальність, рівень вищої освіти (освітньо-кваліфікаційний рівень)	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
Кількість кредитів – 5	Галузь знань 11 – “Математика і статистика”	Нормативна	
		Цикл професійної підготовки	
Розділів – 2	Спеціальність: 113 – “Прикладна математика”	<b>Рік підготовки:</b>	
Загальна кількість годин – 150		3-й	–
		<b>Лекції</b>	
Тижневих годин для денної форми навчання: аудиторних – 4,5 самостійної роботи студента – 6	Освітньо-професійна програма “Прикладна математика”	26 год.	–
		<b>Лабораторні роботи</b>	
	38 год.	–	
	<b>Самостійна робота</b>		
	86 год.	–	
	Рівень вищої освіти: <b>бакалаврський</b>	<b>Вид контролю: екзамен</b>	

## 2. Мета та завдання навчальної дисципліни

Метою курсу є надання знань студентам спеціальності “Прикладна математика” про основні сучасні технології програмування.

### Завдання курсу:

– ознайомити студентів з основними поняттями технології програмування й життєвого циклу програмного забезпечення, а також етапами і технологічними процедурами, що застосовуються при створенні програм;

– навчити їх практично використовувати сучасні об’єктно-орієнтовані та візуальні парадигми при програмуванні;

– навчити студентів застосуванню кросплатформної системи програмування Qt Creator з відкритим програмним кодом для об’єктно-орієнтованого, узагальненого, функціонального та візуального програмування.

### У результаті вивчення дисципліни студент повинен знати:

- 1) основні поняття про сучасні технології програмування;
- 2) основи мови програмування C++;
- 3) базові поняття об’єктно-орієнтованого, узагальненого, функціонального та візуального програмування;
- 4) основні можливості стандартної бібліотеки C++;

- 5) базові поняття розробки програмного забезпечення в системі програмування Qt Creator;
- 6) основні елементи кросплатформної бібліотеки Qt.

**вміти:**

- 1) проектувати й розробляти різні види програмного забезпечення на основі об'єктно-орієнтованого, узагальненого, функціонального та візуального підходу із застосуванням мови програмування C++;
- 2) застосовувати стандартну бібліотеку мови програмування C++ при створенні програм;
- 3) використовувати бібліотеку Qt для створення графічного інтерфейсу програм;
- 4) застосовувати прийоми узагальненого програмування на мові C++;
- 5) використовувати систему програмування Qt Creator для візуального програмування графічних інтерфейсів програм.

Згідно з вимогами освітньо-професійної програми студенти повинні досягти таких **компетентностей**:

- здатність застосовувати вибрану програмну платформу та систему програмування для написання програмного коду;
- здатність розв'язувати складні спеціалізовані задачі та проблеми, що виникають при практичному застосуванні інформаційних технологій у професійній діяльності або в процесі навчання, які, насамперед, передбачають створення складного програмного забезпечення;
- здатність вести дослідницьку діяльність, включаючи аналіз проблем, вибір способу і методів дослідження, а також оцінку якості результатів;
- здатність використовувати мову програмування C++ та об'єктно-орієнтовану технологію програмування для створення програмного забезпечення;
- здатність аргументувати вибір методів розв'язання спеціалізованих задач, критично оцінювати отримані результати, обґрунтовувати та захищати прийняті рішення;
- здатність вирішувати проблеми у галузі комп'ютерних та інформаційних технологій, визначати обмеження цих технологій.

### 3. Міждисциплінарні зв'язки

Для розуміння змісту курсу студент повинен мати знання основ наступних дисциплін:

1. Теорія алгоритмів та програмування.
2. Системне програмування та архітектура комп'ютерів.
3. Об'єктно-орієнтоване програмування.
4. Операційні системи.

У свою чергу, теоретичні відомості та практичні навички, що надаються у курсі “Технології програмного забезпечення” використовуються при вивченні спецкурсів, що викладаються для спеціальності “Інформаційні системи та

технології” згідно навчального плану, виконанні курсових, дипломних та магістерських робіт.

## **4. Програма навчальної дисципліни**

### **Розділ 1. Об’єктно-орієнтоване програмування і мова C++.**

#### **Тема 1. Основи технології програмування.**

Історія створення обчислювальної техніки та розвитку програмування. Основні поняття технології програмування та етапи створення програмного забезпечення. Сучасні парадигми та технології програмування.

#### **Тема 2. Основи об’єктно-орієнтованого програмування на мові C++.**

Загальні поняття про об’єктно-орієнтоване програмування. Класи та об’єкти. Інкапсуляція. Поліморфізм. Наслідування. Перевантаження операцій. Системи програмування на мові C++.

#### **Тема 3. Введення-виведення в мові C++.**

Стандартні потокові класи вводу-виводу. Форматоване введення-виведення. Маніпулятори. Файловий ввід-вивід.

#### **Тема 4. Обробка виняткових ситуацій.**

Загальні поняття про виняткові ситуації. Синтаксис винятків. перехоплення та обробка виняткових ситуацій. Стандартні виняткові ситуації.

#### **Тема 5. Узагальнене програмування. Стандартна бібліотека C++.**

Поняття узагальненого програмування. Шаблони. Шаблони класів і функцій. Еквівалентність шаблонних класів. Строки та строкові потоки. Контейнерні класи. Алгоритми.

### **Розділ 2. Функціональне та візуальне програмування.**

#### **Тема 6. Основи функціонального програмування.**

Поняття функціонального програмування. Лямбда-вирази в C++. Динамічне визначення функцій за допомогою лямбда-виразів. Рекурсія в лямбда-виразах. Приклади застосування лямбда-виразів.

#### **Тема 7. Основи візуального програмування.**

Базові поняття візуального програмування. Сучасні інтегровані середовища візуального програмування.

#### **Тема 8. Бібліотека Qt.**

Функції та склад бібліотеки Qt. Платформи, що підтримуються Qt. Огляд базових компонентів. Застосування.

#### **Тема 9. Створення графічного інтерфейсу засобами Qt.**

Поняття форми та віджету. Основні типи віджетів. Компонування. Сигнали та слоти.

#### **Тема 10. Qt Designer та швидка розробка прототипів інтерфейсів.**

Базові принципи роботи з Qt Designer. Застосування компоувальника. Контейнери. Створення меню. Налаштування форм та віджетів.

## 5. Структура навчальної дисципліни

Назви розділів і тем	Кількість годин										
	денна форма						заочна форма				
	усього	у тому числі					усього	у тому числі			
л		п	лаб	с.р.	інд	л		п	лаб	с.р.	інд
<b>Розділ 1. Об'єктно-орієнтоване програмування і мова C++</b>											
Тема 1. Основи технології програмування.	22	4	0	0	20	0					
Тема 2. Основи об'єктно-орієнтованого програмування на мові C++.	34	4	0	8	22	0					
Тема 3. Введення-виведення в мові C++.	6	2	0	4	0	0					
Тема 4. Обробка виняткових ситуацій.	4	2	0	2	0	0					
Тема 5. Узагальнене програмування на мові C++. Стандартна бібліотека C++.	30	2	0	6	20	0					
<b>Разом за розділом 1</b>	<b>96</b>	<b>14</b>	<b>0</b>	<b>20</b>	<b>62</b>	<b>0</b>					
<b>Розділ 2. Функціональне та візуальне програмування</b>											
Тема 6. Основи функціонального програмування	4	2	0	2	0	0					
Тема 7. Основи візуального програмування.	20	4	0	4	12	0					
Тема 8. Бібліотека Qt.	18	2	0	4	12	0					
Тема 9. Створення графічного інтерфейсу засобами Qt.	6	2	0	4	0	0					
Тема 10. Qt Designer та швидка розробка прототипів інтерфейсів.	6	2	0	4	0	0					
<b>Разом за розділом 2</b>	<b>54</b>	<b>12</b>	<b>0</b>	<b>18</b>	<b>24</b>	<b>0</b>					
<b>Усього годин</b>	<b>150</b>	<b>26</b>	<b>0</b>	<b>38</b>	<b>86</b>	<b>0</b>					

## 6. Теми лекційних занять

№ теми	Назва	Кількість годин
1	<b>Основи технології програмування.</b> Вступ. Історія створення обчислювальної техніки. Основні поняття технології програмування та етапи створення програмного забезпечення. Сучасні парадигми та технології програмування.	4
2	<b>Основи об'єктно-орієнтованого програмування на мові С++.</b> Загальні поняття про об'єктно-орієнтоване програмування. Класи та об'єкти. Інкапсуляція. Поліморфізм. Наслідування. Перевантаження операцій. Системи програмування на мові С++.	4
3	<b>Введення-виведення в мові С++.</b> Стандартні потокові класи вводу-виводу. Форматоване введення-виведення. Маніпулятори. Файловий ввід-вивід.	2
4	<b>Обробка виняткових ситуацій.</b> Загальні поняття про виняткові ситуації. Синтаксис винятків. перехоплення та обробка виняткових ситуацій.	2
5	<b>Узагальнене програмування. Стандартна бібліотека С++.</b> Поняття узагальненого програмування. Шаблони. Шаблони класів і функцій. Еквівалентність шаблонних класів. Строки та строкові потоки. Контейнерні класи. Алгоритми.	2
6	<b>Основи функціонального програмування.</b> Поняття функціонального програмування. Лямбда-вирази в С++. Динамічне визначення функцій за допомогою лямбда-виразів. Рекурсія в лямбда-виразах. Приклади застосування лямбда-виразів.	2
7	<b>Основи візуального програмування.</b> Базові поняття візуального програмування. Сучасні інтегровані середовища візуального програмування.	4
8	<b>Бібліотека Qt.</b> Функції та склад Qt. Платформи, що підтримуються Qt. Огляд базових компонентів. Застосування.	2
9	<b>Створення графічного інтерфейсу засобами Qt.</b> Поняття форми та віджету. Основні типи віджетів. Компонування. Сигнали та слоти.	2
10	<b>Qt Designer та швидка розробка прототипів інтерфейсів.</b> Базові принципи роботи з Qt Designer. Застосування компоувальника. Контейнери. Створення меню. Налаштування форм та віджетів.	2
<b>Разом</b>		<b>26</b>

## 7. Теми лабораторних занять

№ теми	Назва	Кількість годин
2	<b>Створення програм на мові C++ в Qt Creator.</b> Створення проекту в Qt Creator. Редагування вихідного коду. Компіляція та запуск програм. Застосування налагоджувача.	2
2	<b>Основи об'єктно-орієнтованого програмування в C++.</b> Створення класів і об'єктів в C++. Конструктори. Деструктори. Перевантаження операцій. Інкапсуляція. Поліморфізм. Наслідування.	8
3	<b>Введення-виведення в C++.</b> Потокові класи вводу-виводу. Стандартні потоки. Маніпуляції з потоками.	2
3	<b>Файлове введення-виведення.</b> Класи ifstream та ofstream.	2
4	<b>Обробка виняткових ситуацій.</b> Конструкція try...catch. Створення власного обробника винятків. Стандартні класи обробки виняткових ситуацій в C++.	2
5	<b>Узагальнене програмування. Стандартна бібліотека C++.</b> Створення шаблонних класів та функцій. Робота зі строками та строковими потоками. Стандартні контейнери бібліотеки STL. Алгоритми.	6
7	<b>Функціональне програмування в C++.</b> Поняття функціонального програмування. Лямбда-вирази в C++. Приклад функціональної програми на C++.	2
8	<b>Бібліотека Qt.</b> Консольний проект в Qt. Строки та контейнери. Потоки.	4
9	<b>Створення графічного інтерфейсу засобами Qt.</b> Створення програм з графічним інтерфейсом в Qt. Головне та діалогові вікна. Меню. Базові віджети. Сигнали та слоти.	4
10	<b>Qt Designer та швидка розробка прототипів інтерфейсів.</b> Базові принципи роботи з Qt Designer. Застосування компоувальника. Контейнери. Створення меню. Налаштування форм та віджетів.	4
<b>Разом</b>		<b>38</b>

## 8. Самостійна робота

№ теми	Назва	Кількість годин
1	Процедурна та структурна технології програмування	20
2	Стандарти мови програмування C++	12
2	Лямбда-вирази та елементи функціонального програмування в C++	10
5	Стандартна бібліотека STL	20
7	Створення багатопотокових програм із застосуванням Qt	12
8	Створення графічного інтерфейсу засобами Qt.	12
<b>Разом</b>		<b>86</b>

## 9. Види контролю і система накопичення балів

### Методи контролю

Кожна лабораторна робота складається з теоретичної і практичної частини. Теоретична частина захищається студентом шляхом бесіди з викладачем. Практична частина перевіряється викладачем на предмет якості написання програмного коду та відповідності результатів роботи програми заявленим задачам.

### Розподіл балів, які отримують студенти

Розділ	Вид контролю	Кількість балів
<b>Розділ 1</b>	Лабораторна робота № 1	6
	Лабораторна робота № 2	6
	Лабораторна робота № 3	6
	Лабораторна робота № 4	6
	Лабораторна робота № 5	6
<b>Разом</b>		<b>30</b>
<b>Розділ 2</b>	Лабораторна робота № 6	6
	Лабораторна робота № 7	6
	Лабораторна робота № 8	6
	Лабораторна робота № 9	6
	Лабораторна робота № 10	6
<b>Разом</b>		<b>60</b>
<b>Екзамен</b>		<b>40</b>
<b>Всього за семестр</b>		<b>100</b>

### Критерії оцінювання кожного з проведених видів контролю:

1. Захист лабораторної роботи відбувається після виконання завдання та завантаження відповідного звіту до системи Moodle. Один завантажений звіт без захисту лабораторної роботи може бути оцінений не вище, ніж 50% від максимального можливого значення балів. Під час захисту студент має відповісти на питання викладача та пояснити деякі етапи виконання завдання.

2. Екзамен проводиться у формі тестування рівня теоретичної підготовки з усіх розділів курсу та має 40 питань, які оцінюються по 1 балу.

3. Підсумкова оцінка складається з суми балів, отриманих студентом за виконання лабораторних робіт, та оцінки, отриманої на екзамені.



### Шкала оцінювання: національна та ECTS

За шкалою ECTS	За шкалою університету	За національною шкалою	
		Екзамен	Залік
A	90 – 100 (відмінно)	5 (відмінно)	Зараховано
B	85 – 89 (дуже добре)	4 (добре)	
C	75 – 84 (добре)		
D	70 – 74 (задовільно)	3 (задовільно)	
E	60 – 69 (достатньо)		
FX	35 – 59 (незадовільно – з можливістю повторного складання)	2 (незадовільно)	Не зараховано
F	1 – 34 (незадовільно – з обов’язковим повторним курсом)		

## 10. Рекомендована література

### Основна

1. Камаев В. А. Технологии программирования: Учебник. – М. Высш. шк., 2006. – 454 с.
2. Страуструп Б. Язык программирования С++. Специальное издание. Пер. с англ. – М. : Издательство Бином, 2011. – 1136 с.
3. Скотт М. Эффективный и современный С++: 42 рекомендации по использованию С++ 11 и С++14.: Пер. с англ. – М.: ООО "ИЛ. Вильяме", 2016. – 304 с.
4. Прата С. Язык программирования С. Лекции и упражнения, 6-е изд. : Пер. с англ. – М. : Вильямс, 2015. – 928 с.
5. Лафоре Р. Объектно-ориентированное программирование в С++. 4-е издание. – М. Питер, 2004. – 923 с.
6. Галовиц Я. С++17 STL. Стандартная библиотека шаблонов. — СПб.: Питер, 2018. — 432 с.

### Додаткова:

7. Программирование на языке С++ в среде Qt Creator: / Е.Р. Алексеев, Г.Г. Злобин, Д.А. Костюк, О.В. Чеснокова, А.С. Чмыхало – М. : ALT Linux, 2015. – 448с.
8. Бланшет Ж., Саммерфилд М. Qt 4: программирование GUI на С++. – М. : КУДИЦ-ПРЕСС, 2008. – 736 с.
9. Шлее М. Qt 4.5. Профессиональное программирование на С++. – СПб. : БХВ-Петербург, 2010. – 896 с.
10. Шлеер С., Меллор С. Объектно-ориентированный анализ: моделирование мира в состояниях. – К. : Диалектика, 1993. – 240 с.
11. Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений – М. : Вильямс, 2008. – 720 с.

Погоджено \_\_\_\_\_  
 навчальний відділ  
 “ ” \_\_\_\_\_ 2019 р.