



ОБ'ЄКТНО-ОРІЄНТОВАНЕ ПРОГРАМУВАННЯ НА ОСНОВІ ТЕХНОЛОГІЇ QT

Робоча програма навчальної дисципліни (Силабус)

Реквізити навчальної дисципліни

Рівень вищої освіти	Перший (бакалаврський)
Галузь знань	17 — Електроніка, автоматизація та електронні комунікації
Спеціальність	172 – Електронні комунікації та радіотехніка
Освітня програма	Інтелектуальні технології радіоелектронної техніки Інформаційна та комунікаційна радіоінженерія Радіотехнічні комп'ютеризовані системи
Статус дисципліни	Вибіркова
Форма навчання	очна(денна)
Рік підготовки, семестр	2 курс, 3 семестр
Обсяг дисципліни	120 годин (18 годин – Лекції, 36 годин – лабораторні роботи, 66 годин – СРС)
Семестровий контроль/ контрольні заходи	Залік/залікова робота, МКР
Розклад занять	
Мова викладання	Українська
Інформація про керівника курсу / викладачів	Лектор: к.т.н., доцент Катін Павло Юрійович, katin.dino@gmail.com , моб. +38(098)202-08-11 Лабораторні: к.т.н., доцент Катін Павло Юрійович,
Розміщення курсу	

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати. Метою вивчення дисципліни (кредитного модуля) «Об'єктно-орієнтоване програмування на основі технології Qt» є надання студентам знань і навичок для самостійної розробки архітектури і програмної реалізації прототипів, що можуть стати основою радіотехнічних систем та інших програмних рішень.

Синтаксис C++ у контексті використання мови у системі бібліотек Qt, технологію розробки, налагодження і підтримки програмного забезпечення. Основи об'єктно-орієнтованого програмування (ООП) та особливості парадигми ООП у програмах на основі Qt. Різні варіанти архітектур програмних рішень на базі C++ у контексті використання мови у системі бібліотек Qt.

Вивчення дисципліни (кредитний модуль) дозволяє:

створювати графічні інтерфейси користувача з використанням бібліотек Qt та мови програмування C++;

створювати програми для роботи із реляційними базами даних з використанням бібліотек Qt та мови програмування C++;

створювати програми для роботи із драйверами пристроїв периферії, файлів та мережевих технологій;

забезпечувати платформозалежності отриманих програмних рішень.

2. Пререквізити та постреквізити дисципліни (місце в структурно-логічній схемі навчання за відповідною освітньою програмою)

1. Успішне вивчення дисципліни (кредитного модуля) ґрунтується на дисципліні «Програмування» навчального плану підготовки бакалаврів спеціальності 172.

Отримані під час засвоєння дисципліни (залікового модуля) теоретичні знання та практичні навички необхідні для розробки бакалаврського проекту.

3. Зміст навчальної дисципліни

Тема 1. Синтаксис C++ у Qt – базові конструкції мови, сигнали та слоти, метаоб'єкти.

Тема 2. Розробка та налагодження – Qt Creator, CMake/qmake, профілювання, тестування.

Тема 3. ООП у Qt – інкапсуляція, наслідування, поліморфізм, QObject та QWidget.

Тема 4. Архітектури – MVC, MVVM, патерни проектування, сигнально-слотова модель.

Тема 5. GUI – Qt Widgets, Qt Quick, QML, створення інтерфейсів.

Тема 6. Бази даних – Qt SQL, SQLite/MySQL/PostgreSQL, CRUD-операції.

Тема 7. Драйвери, файли, мережі – QFile, QIODevice, TCP/UDP клієнт-сервер.

Тема 8. Платформозалежність – кросплатформеність Qt, умовна компіляція, тестування на різних ОС.

Тема 9. Підсумковий огляд – інтеграція знань, розробка комплексного застосунку.

Модульна контрольна робота залік

4. Навчальні матеріали та ресурси

Основна література:

1. Страуструп, Б. The C++ Programming Language. 4-е вид. – Boston : Addison-Wesley, 2013. – 1376 с.
2. Blanchette, J., Summerfield, M. C++ GUI Programming with Qt 4. – Upper Saddle River : Prentice Hall, 2006. – 512 с.
3. Summerfield, M. Advanced Qt Programming: Creating Great Software with C++ and Qt 4. – Upper Saddle River : Addison-Wesley, 2010. – 528 с.

Додаткова література:

1. Blanchette, J., Summerfield, M. C++ GUI Programming with Qt 5. – Upper Saddle River : Prentice Hall, 2016. – 624 с.
2. Meyers, S. Effective C++: 55 Specific Ways to Improve Your Programs and Designs. 3-е изд. – Boston : Addison-Wesley, 2005. – 320 с.

Навчальний контент

1. Методика опанування навчальної дисципліни (освітнього компонента).

№.	Тип занять	Опис змісту заняття
<i>Тема 1. Синтаксис C++ у Qt – базові конструкції мови, сигнали та слоти, метаоб'єкти.</i>		
1	Лекція 1. Основи синтаксису C++ та інтеграція з Qt	Розглядаються ключові конструкції C++ (класи, шаблони, винятки). Пояснюється роль препроцесора, простори імен та стандартна бібліотека. Демонструється механізм сигналів і слотів у Qt як розширення C++. Показується приклад простої програми на Qt з використанням базових елементів.
2		
3	<i>Лабораторна робота 1.</i>	<i>Ознайомлення з синтаксисом C++ та створення першої програми на Qt</i>
<i>Тема 2. Розробка та налагодження – Qt Creator, CMake/qmake, профілювання, тестування.</i>		
4	<i>Лекція 2. Інструменти розробки та налагодження у Qt Creator</i>	ТОгляд середовища Qt Creator, систем збірки qmake та CMake. Методи налагодження програм: точки зупину, перегляд змінних, профілювання. Використання unit-тестів для перевірки функціональності. Практичні аспекти підтримки та документування коду.
5		
6	<i>Лабораторна робота 2.</i>	<i>Використання сигналів і слотів у Qt</i>
<i>Тема 3. ООП у Qt – інкапсуляція, наслідування, поліморфізм, QObject та QWidget.</i>		
7	<i>Лекція 3. Об'єктно-орієнтоване програмування та Qt-парадигма</i>	Інкапсуляція, наслідування та поліморфізм у C++. Особливості класу QObject та його роль у Qt. Використання віртуальних методів та перевизначення функцій. Приклади створення власних віджетів на основі базових класів Qt.
8		
9	<i>Лабораторна робота 3.</i>	<i>Реалізація класів та наслідування у програмах Qt</i>
<i>Тема 4. Архітектури – MVC, MVVM, патерни проектування, сигнально-слотова модель.</i>		
10	<i>Лекція 4. Архітектурні підходи у C++/Qt</i>	Огляд моделей MVC та MVVM у контексті Qt. Сигнально-слотова архітектура як основа взаємодії компонентів. Використання патернів проектування (Observer, Singleton, Factory).
11		

		Приклади побудови невеликих систем із застосуванням архітектурних рішень.
12	Лабораторна робота 4.	Побудова архітектури застосунку (MVC/MVVM)
Midterm Test		
Тема 5. GUI – Qt Widgets, Qt Quick, QML, створення інтерфейсів.		
13	Лекція 5. Створення графічних інтерфейсів	Основи роботи з Qt Widgets: кнопки, меню, діалоги. Використання Qt Designer для швидкого створення інтерфейсів. Вступ до QML та його роль у сучасних UI. Приклади комбінування QML та C++ у одному застосунку.
14		
15	Лабораторна робота 5.	Створення графічного інтерфейсу користувача з Qt Widgets
Тема 6. Бази даних – Qt SQL, SQLite/MySQL/PostgreSQL, CRUD-операції.		
16	Лекція 6. Qt SQL та інтеграція з базами даних	Огляд Qt SQL Module та його можливостей. Підключення до SQLite, MySQL та PostgreSQL. Виконання CRUD-операцій через Qt. Приклади створення застосунку з базою даних.
17		
18	Лабораторна робота 6.	Робота з базами даних у Qt SQL (CRUD-операції)
Тема 7. Драйвери, файли, мережі – QFile, QIODevice, TCP/UDP клієнт-сервер.		
19	Лекція 7. Qt Network та файлові операції	Використання класів QFile та QIODevice для роботи з файлами. Основи роботи з мережевими протоколами TCP та UDP. Створення клієнт-серверних застосунків у Qt. Приклади взаємодії з периферійними пристроями.
20		
21	Лабораторна робота 7.	Робота з файлами та мережевими технологіями у Qt
Тема 8. Платформозалежність – кросплатформеність Qt, умовна компіляція, тестування на різних ОС.		
22	Лекція 8. Кросплатформеність та умовна компіляція .	Особливості роботи Qt на Windows, Linux та macOS. Використання умовної компіляції (#ifdef, #endif). Тестування програм на різних операційних системах. Практичні приклади адаптації коду під різні платформи.
23		
24	Лабораторна робота 8.	Кросплатформене тестування та адаптація програмного забезпечення
Тема 9. Підсумковий огляд – інтеграція знань, розробка комплексного застосунку.		

25	Лекція 9. Комплексна розробка застосунку на C++/Qt	Узагальнення знань з попередніх модулів. Проектування комплексного застосунку з GUI, базою даних та мережевими можливостями. Обговорення типових проблем та шляхів їх вирішення. Підготовка до захисту курсового проєкту.
26		
Залік		

6. Самостійна робота студента/аспіранта

Дисципліна ґрунтується на самостійних підготовках до аудиторних занять на теоретичні та практичні теми.

№	Напрямок самостійної підготовки	Кількість годин	Література
1	Підготовка до лекції 1	1	1-5
2	Підготовка до лабораторної роботи 1	1,5	1-5
3	Підготовка до лекції 1	1	1-5
4	Підготовка до лекції 2	1	1-5
5	Підготовка до лабораторної роботи 2	1,5	1-5
6	Підготовка до лекції 2	1	1-5
7	Підготовка до лекції 3	1	1-5
8	Підготовка до лабораторної роботи 3 (part 1)	1,5	1-5
9	Підготовка до лекції 3	1	1-5
10	Підготовка до лекції 4	1	1-5
11	Підготовка до лабораторної роботи 3 (part 2)	1,5	1-5
12	Підготовка до лекції 4	1	1-5
13	Підготовка до лекції 5	1	1-5
14	Підготовка до лабораторної роботи 4 (part 1)	1,5	1-5
15	Підготовка до лекції 5	1	1-5
16	Підготовка до лекції 6	1	1-5
17	Підготовка до лабораторної роботи 4 (part 2)	1,5	
18	Підготовка до лекції 7	1	
19	Підготовка до лекції 7	1	
20	Підготовка до лабораторної роботи 5	1,5	
21	Підготовка до лекції 8	1	1-5
22	Підготовка до лекції 8	1	1-5

23	Підготовка до лабораторної роботи 6 (part 1)	1,5	1-5
24	Підготовка до лекції 16	1	1-5
25	Підготовка до лекції 9	1	1-5
26	Підготовка до лабораторної роботи 6 (part 2)	1,5	1-5
27	Preparation for the midterm test	4	1-5
28	Qt SQL Module	30	1-5
29	MVC та MVVM у контексті Qt.	2	1
30	SQLite, MySQL та PostgreSQL	1	1
31	Essential C++ Features for web developing. Stage 1.	1	1
32	Essential C++ Features for web developing. Stage 1.	1	1
33	Essential C++ Features for web developing. Stage 2.	1,5	1
34	Essential C++ Features for web developing. Stage 2.	1	1
35	Essential C++ Features for web developing. Stage 3.	1	1
36	Essential C++ Features for web developing. Stage 3.	2	1
37	Essential C++ Features for web developing. Stage 4.	2	1
38	Essential C++ Features for web developing. Stage 4.	1	1
39	QFile та QIODevice	1	1
40	TCP та UDP.	2	1

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

Відвідування лекційних занять є обов'язковим. В умовах особливого стану питання щодо відвідування лекційних занять і інших аспектів політики може бути змінено.

Відвідування занять комп'ютерного практикуму може бути епізодичним та за потреби консультації/захисту робіт лабораторних робіт.

Правила поведінки на заняттях: активність, повага до присутніх, відключення телефонів.

Дотримання політики академічної доброчесності.

Правила захисту лабораторних робіт: роботи повинні бути зроблені відповідно до поставлених задач та згідно з варіантом.

8. Види контролю та рейтингова система оцінювання результатів навчання (PCO)

Протягом семестру студенти виконують 8 лабораторних робіт. Максимальна кількість балів за кожну: 10 балів.

Бали нараховуються за:

- якість виконання лабораторних робіт: 0-5 бали;
- відповідь під час захисту лабораторних робіт: 0-3 бали;
- своєчасне представлення роботи до захисту: 0-2 бали.

Критерії оцінювання якості виконання:

5 бали – робота виконана якісно, в повному обсязі;

4 бали – робота виконана якісно, в повному обсязі, але має недоліки;
3 бали – робота виконана в повному обсязі, але містить незначні помилки;
2 бали – робота виконана в повному обсязі, але містить суттєві помилки;
0 балів – робота виконана не в повному обсязі.

Критерії оцінювання відповіді:

3 бали – відповідь повна, добре аргументована;
2 бали – відповідь вірна, але має недоліки або незначні помилки;
1 бал – у відповіді є суттєві помилки;
0 балів – немає відповіді або відповідь невірна.

Критерії оцінювання своєчасності представлення роботи до захисту:

2 бали – робота представлена до захисту не пізніше вказаного терміну;
0 балів – робота представлена до захисту пізніше вказаного терміну.

Максимальна кількість балів за виконання та захист лабораторних робіт:

10 балів × 8 л.р. = 80 балів.

Протягом семестру на лекціях відбуваються опитування за темою поточного заняття. Максимальна кількість балів за всі опитування: 3 бали. Кількість опитування за темою поточного заняття для одного студента є необмеженою.

Завдання на модульну контрольну роботу складається з 1 теоретичних та 1 практичних запитань. Відповідь оцінюється максимально 20 балами.

Критерії оцінювання кожного запитання контрольної роботи:

9-10 балів – відповідь вірна, повна, добре аргументована;
7-8 балів – відповідь вірна, розгорнута, але не дуже добре аргументована;
5-6 балів – в цілому відповідь вірна, але має недоліки;
3-4 балів – у відповіді є незначні помилки;
1-2 бали – у відповіді є суттєві помилки;
0 балів – немає відповіді або відповідь невірна.

Максимальна кількість балів за модульну контрольну роботу:

10 балів × 2 запитань = 20 балів.

Рейтингова шкала з дисципліни дорівнює:

$R = R_C = 80 \text{ балів} + 20 \text{ балів} = 100 \text{ балів}.$

Календарний контроль: проводиться двічі на семестр як моніторинг поточного стану виконання вимог силабусу.

На першій атестації (8-й тиждень) студент отримує «зараховано», якщо його поточний рейтинг не менше 15 балів (50 % від максимальної кількості балів, яку може отримати студент до першої атестації).

На другій атестації (14-й тиждень) студент отримує «зараховано», якщо його поточний рейтинг не менше 20 балів (50 % від максимальної кількості балів, яку може отримати студент до другої атестації).

Семестровий контроль: залік

Умови допуску до семестрового контролю:

При семестровому рейтингу (R_C) не менше 60 балів та зарахуванні усіх робіт комп'ютерного практикуму, студент отримує залік «автоматом» відповідно до таблиці (Таблиця відповідності рейтингових балів оцінкам за університетською шкалою). В іншому разі він має виконувати залікову контрольну роботу.

Необхідною умовою допуску до залікової контрольної роботи є виконання і захист лабораторних робіт.

Якщо студент не погоджується з оцінкою «автоматом», то може спробувати підвищити свою оцінку шляхом написання залікової контрольної роботи, при цьому його бали, отримані за семестр, зберігаються, а з двох отриманих студентом оцінок виставляється краща («м'яка» система оцінювання).

Таблиця відповідності рейтингових балів оцінкам за університетською шкалою:

Кількість балів	Оцінка
100-95	Відмінно
94-85	Дуже добре
84-75	Добре
74-65	Задовільно
64-60	Достатньо
Менше 60	Незадовільно
Не виконані умови допуску	Не допущено

9. Додаткова інформація з дисципліни (освітнього компонента)

Для роботи можуть бути використані різні операційні системи за бажанням студента і по узгодженню з викладачем.

У випадку бажання студента використовувати у комп'ютерному практикумі інші мови програмування або технології вони можуть бути включені як додатковий елемент у базову архітектуру розподіленого веб-застосунку.

Робочу програму навчальної дисципліни (силабус):

Складено доцент кафедри, к.т.н., доцент, Катін П.Ю.

Ухвалено кафедрою РТС_ (протокол № 06/2024_ від _ 27.06. 2024 р.)

Погоджено Методичною комісією радіотехнічного факультету (протокол № _ 06/2024 _ від 28.06.2024 р.)