

[RE-61] ТЕХНОЛОГІЇ ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ ДЛЯ РАДІОІНЖЕНЕРІВ



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	Всі ОП
Статус дисципліни	Вибіркова (Ф-каталог)
Форма здобуття вищої освіти	Очна
Рік підготовки, семестр	Доступно для вибору починаючи з 2-го курсу, весняний семестр
Обсяг дисципліни	4 кред. (Лекц. 18 год, Практик. год, Лаб. 36 год, СРС. 66 год)
Семестровий контроль/контрольні заходи	Залік
Розклад занять	https://schedule.kpi.ua
Мова викладання	Українська / Англійська
Інформація про керівника курсу / викладачів	Лекц.: Неуймін О. С. , Лаб.: Неуймін О. С. , СРС.: Неуймін О. С.
Розміщення курсу	https://do.ipk.kpi.ua/course/view.php?id=2521

Програма навчальної дисципліни

1. Опис навчальної дисципліни, її мета, предмет вивчення та результати навчання

Об'єктно-орієнтоване програмування на C++

C++ створений на базі мови C, і на даний момент підтримує різні стилі програмування, у тому числі об'єктно-орієнтований стиль, що дозволяє з легкістю підтримувати великі проекти. Ця мова залишається одна з найшвидкісних, і використовується в різноманітних сферах

застосування - від графічних інтерфейсів додатків та ігрової 3D-графіки до робототехніки. C++ використовується всіма великими компаніями, наприклад *Amazon, Google, Microsoft*, а в Україні: *GlobalLogic, Samsung, Luxoft, Infopulse*. Також мова активно розвивається, оскільки кожних три роки приймається новий стандарт спеціальним комітетом по стандартизації C++, в якому додаються все нові і нові можливості мови. Програмісти C++ затребувані у всьому світі, про що свідчить кількість відкритих вакансій та рівень заробітних плат.

Метою курсу є формування у студентів здатностей:

- застосовувати основи технології об'єктно-орієнтованого програмування та базові патерни проектування при створенні програмного забезпечення для радіотехнічних інформаційних систем;
- створювати програмне забезпечення із відповідним функціоналом для радіоелектронної апаратури;
- відлагоджувати та реалізовувати програми в різних середовищах програмування;
- знаходити, оцінювати й використовувати інформацію з різних джерел, необхідну для рішення професійних завдань;
- працювати в команді.

Студенти отримають знання з:

- основ технології об'єктно-орієнтованого програмування;
- базових патернів проектування;
- відносин між класами і основи UML (діаграми класів і послідовностей);
- основних інструментальних засобів мови C++ і стандартної бібліотеки STL;
- системи контролю версій Git.

Студенти зможуть:

- застосовувати об'єктно-орієнтований підхід під час проектування складних програмних систем із відповідним функціоналом;
- реалізувати програмне забезпечення з урахуванням вимог до його якості, надійності, виробничих характеристик;
- володіти інструментальними засобами мови C++;
- реалізовувати та відлагоджувати програми в різних середовищах програмування.

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

Дисципліни, які забезпечують «Об'єктно-орієнтоване програмування для радіоінженерів» є: **«Вища математика»**, **«Загальна фізика»**, **«Інформатика»**. Для успішного засвоєння дисципліни студент повинен мати базовий рівень володіння мовою програмування C.

Дисципліна є основою для вивчення всіх наступних дисциплін, в яких необхідно використовувати принципи об'єктно-орієнтованого програмування для написання програмного забезпечення, наприклад Дипломний проєкт.

3. Зміст навчальної дисципліни

Основні теми дисципліни

1. Робота з пам'яттю. Інтерпретація пам'яті.
2. Статичні локальні змінні в функції. Вказівники на функції.
3. Основні концепції об'єктно-орієнтованого програмування: **інкапсуляція, поліморфізм, успадкування**.
4. Класи. Конструктори. Деструктори. Правила ініціалізації. Перевантажені конструктори.

Конструктор копіювання і перевантаження оператора присвоювання. Вказівник **this**. Вказівники на члени класу і їх застосування.

5. Перевантаження операторів. Правила. Розуміння ситуацій, коли необхідна перевантаження. Діагностика операторів, які виконуються при перевантаженні. Перевантаження операторів **+**, **=**, **[]**, **++**, **()**, **приведення типу**, **->**, **new**. **delete**. **placement new**.
6. Дружні функції, дружні класи, попереднє оголошення класу. Перевантаження операторів "друзів".
7. Статичні члени класу (статичні змінні-члени, методи-члени), патерн **Singleton**.
8. Використання модифікаторів **const**, **mutable**. Зняття константності (модифікатор **mutable**, **const_cast**).
9. Динамічні структури даних. Однозв'язний список. Двухзв'язний список. Черга. Кільцева черга. Черга з пріоритетами. Бінарне дерево.
10. Робота зі стандартною бібліотекою C++ (**STL**). Контейнери. Ітератори.
11. Алгоритми (правила використання; плюси і мінуси) **big O notation**.
12. Спадкування. Правила ініціалізації при спадкуванні. Одиночне спадкоємство, множинне спадкування, віртуальне успадкування.
13. Віртуальні функції. Раннє і пізніше зв'язування. Необхідність віртуального деструктора.
14. Абстрактні класи та Інтерфейси.
15. Патерни проектування. Агрегування і обізнаність. **UML** діаграми класів, взаємодії об'єктів. Породжувальні патерни. Структурні патерни. Патерни поведінки.
16. Обробка виняткових ситуацій. **try**; **catch**; **throw**. Використання **throw** для спрощення логіки і швидкості. Написання класів-оболонок (розумних вказівників).
17. Система контролю версій **Git**.

Додаткові теми дисципліни

1. Робота з файлами та каталогами в мові C++.
2. Простір імен.
3. Шаблони. Шаблони функцій. Шаблони класів (спеціалізація шаблонів, шаблонні параметри, нетипові параметри шаблонів).
4. Класи бібліотеки **boost**.
5. Динамічні бібліотеки **DLL**.
6. Мережеві додатки. Блокуючі і неблокуючі режими, протоколи.
7. Робота з базами даних, створення баз, таблиць.
8. Робота з фреймворком **Qt**.
9. Функції з невизначеною к-тю параметрів. Формат файлу.

4. Навчальні матеріали та ресурси

Базова література:

1. Васильєв, О. Програмування на C++ в прикладах і задачах : навч. посібник / О. Васильєв. — Київ : Ліра-К, 2017. — 382 с.
2. Програмування мовами C та C++: навч. посіб. / Д.Д. Татарчук, Ю.В. Діденко. – К.: , 2012. –112 с.. [Електронний ресурс]. Режим доступу: https://ela.kpi.ua/bitstream/123456789/25787/1/NP_PM_C_ta_C%2B%2B.pdf
3. Ерік Фрімен. Head First. Патерни проектування. Харків: Фабула, 2020.
4. Неуймін, О. С. Технології об'єктно-орієнтованого програмування для радіоінженерів. Домашня контрольна робота [Електронний ресурс] : навчальний посібник для здобувачів ступеня бакалавра за освітніми програмами «Радіотехнічні комп'ютеризовані системи», «Інформаційна та комунікаційна радіоінженерія», «Інтелектуальні технології радіоелектронної техніки» спеціальності 172 «Телекомунікації та радіотехніка» 172 «Електронні комунікації та радіотехніка» / О. С. Неуймін, О. Ю. Мирончук ; КПІ ім. Ігоря Сікорського. – Електронні текстові дані (1 файл: 1 Мбайт). – Київ : КПІ ім. Ігоря

Додаткова література:

1. Stephen Prata. C++ Primer Plus. Addison-Wesley Professional; 6th edition. – 1440 P.
2. Scott Chacon, Ben Straub. Pro Git book. Посилання: <https://git-scm.com/book/uk/v2>
3. Nicolai M. Josuttis. The C++ Standard Library - A Tutorial and Reference, 2nd Edition. Addison Wesley Longman, 2012.
4. Anthony Williams. C++ Concurrency in Action: Practical Multithreading. Manning; 1st edition, 2012 – 528 P.

Деякі Книги можна знайти за посиланням - <https://ua.booksee.org/>

Телеграм товариство по C++ в КПІ ім. І. Сікорського - https://t.me/itkpi_cpp

Навчальний контент

5. Методика опанування навчальної дисципліни (освітнього компонента)

- Лекції проводяться згідно Тем наведених в розділі 3 «Зміст навчальної дисципліни».
- Практичні заняття навчальним планом дисципліни не передбачені.
- Семінарські заняття навчальним планом дисципліни не передбачені.
- Лабораторні роботи виконуються в системі GitHub Classroom або Moodle.

Теми лабораторних робіт:

1. «Основи використання ООП мови C++». Мета роботи – ознайомитися з основними принципами об'єктно-орієнтованого програмування.
2. «Перевантаження операцій».
3. «Застосування спадкування та поліморфізму».
4. «Динамічні структури даних»
5. «Робота зі стандартною бібліотекою C++ (STL)»
6. «Обробка виняткових ситуацій»

ЛР №1. Реалізація Array

ЛР №2. Реалізація Стека (Stack)

ЛР №3. Завантаження проекту на GitHub

ЛР №4. Перевантаження операторів для класу Array

ЛР №5. Двонаправлений зв'язаний список

6. Самостійна робота студента

Назва теми, що виноситься на самостійне опрацювання

1. Робота з файлами та каталогами в мові C++.
2. Простір імен.
3. Шаблони. Шаблони функцій. Шаблони класів (спеціалізація шаблонів, шаблонні параметри, нетипові параметри шаблонів).
4. Класи бібліотеки Boost.
5. Динамічні бібліотеки DLL.
6. Мережеві додатки. Блокуючі і неблокуючі режими, протоколи.
7. Робота з базами даних, створення баз, таблиць.
8. Робота з фреймворком Qt.
9. Функції з невизначеною кількістю параметрів. Формат файлу.

Домашня контрольна робота

Мета домашньої контрольної роботи – більш глибоке засвоєння матеріалів теоретичного курсу, закріплення навиків самостійного використання набутих знань. Робота проводиться за всіма темами.

Завдання повинне бути виконане відповідно до одного з варіантів. Вибір варіанта для кожного завдання здійснюється відповідно до номера студента в журналі групи.

Виконуючи кожне завдання, спочатку необхідно розробити діаграму класів з використанням мови UML, а потім написати код програми на мові програмування C++ з використанням середовища розробки, наприклад, Visual Studio.

Вимоги до ДКР

- Проєкт має використовувати хоча б одну сторонню бібліотеку.
- Програма складається з мінімум двох класів.
- Проєкт має мати UML діаграму.
- Кінцевий результат проєкту має знаходитися на GitHub.

Приклад завдання

Створити класи, специфікації яких наведені нижче. Визначити конструктори і методи set.. (), get.. (), toString (). Задати критерій вибору даних і вивести ці дані на консоль або вікно GUI. У кожному класі, що володіє інформацією, має бути оголошено кілька конструкторів. Необхідно перевантажити оператор '«' для виводу даних на консоль або вікно GUI. Має бути можливість записати дані всіх об'єктів у файл з шифруванням (студент самостійно обирає спосіб шифрування) або базу даних (БД).

1. **Student:** id, Прізвище, Ім'я, Дата народження, телефон, Факультет, Курс, Група. Створити масив об'єктів. Вивести: а) список студентів заданого факультету; б) списки студентів для кожного факультету і курсу; с) список студентів, які народилися після заданого року; д) список студентів навчальної групи.

Час, на який відводиться виконання ДКР, не менше двох тижнів.

Політика та контроль

7. Політика навчальної дисципліни (освітнього компонента)

- Студент зобов'язаний відвідувати всі лекції та лабораторні заняття.
- За активність на заняттях студенту призначаються заохочувальні бали.
- Студент повинен виконати домашні завдання до встановленої викладачем дати.
- За виявлення плагіату до студента будуть застосовані штрафні бали.
- Вразі невиконання навчального плану і наявності поважних на це причин, студенту може бути видане індивідуальне завдання.

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

Поточний контроль: опитування за темою заняття, МКР, тест.

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

Семестровий контроль: залік / захист ДКР

Умови допуску до семестрового контролю:

- мінімально позитивна оцінка за індивідуальне завдання – 5 балів;

- зарахування усіх лабораторних робіт;
- семестровий рейтинг більше 40 балів.

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)

Приклади питань, які виносяться на семестровий контроль:

1. Основні концепції об'єктно-орієнтованого програмування: інкапсуляція.
2. Основні концепції об'єктно-орієнтованого програмування успадкування.
3. Основні концепції об'єктно-орієнтованого програмування поліморфізм.
4. Правило ініціалізації членів класу.
5. Для чого потрібні Методи, Конструктори.
6. Для чого потрібні Деструктори.
7. Специфікатори доступу.
8. Що таке Перевантаження операторів (приклади: +, =, [], ++, (), приведення типу, ->).
9. Які Правила перевантаження операторів знаєте.
10. Чи можна перевантажувати конструктор?
11. Для чого потрібен Конструктор копіювання
12. Для чого потрібна перевантаження оператора присвоювання.
13. Що таке Дрібне і глибоке копіювання. Вказівник this.
14. Особливості статичних членів класу (статичні змінні-члени, методи-члени).
15. Як відбувається Ініціалізація об'єкта спадкоємця.
16. Які є Відносини між похідним і базовим класами.
17. Що для чого потрібен специфікатор protected
18. Віртуальні функції, деструкція,
19. Що таке абстрактний базовий клас,
20. Що таке інтерфейс
21. Які відмінності між Зв'язаним списком і масивом.

22. Які контейнери STL вам відомі.
23. Що таке ітератор
24. Коли потрібен патерн проектування Singleton.
25. Навіщо потрібен віртуальний деструктор?
26. Для яких цілей застосовується ключове слово const?
27. Як захистити об'єкт від копіювання?
28. У чому різниця між struct і class?
29. У чому відмінність malloc від new?
30. У чому відмінності між delete і delete []?

Можливість зарахування сертифікатів проходження дистанційних чи онлайн курсів за відповідною тематикою;

Студенту може бути зарахований сертифікат проходження онлайн курсів за відповідною тематикою тільки при попередній домовленості з викладачем та затвердження онлайн-курсу за темою ООП на C++.

Оцінка виставляється за результатами співбесіди зі студентом після проходження онлайн курсу.

Опис матеріально-технічного та інформаційного забезпечення дисципліни

Заняття проводяться на комп'ютерах, розміщених в навчальному класі кафедри РТС. Наявна можливість використання власних ноутбуків студентів.

Використовується наступне програмне забезпечення: **Microsoft Visual Studio, Qt Creator, Git, Conan.**

Посилання на платформу дистанційного навчання *Moodle* -
<https://do.ipro.kpi.ua/course/view.php?id=2521>

Робочу програму навчальної дисципліни (силабус):

Складено [Неуймін О. С.](#);

Ухвалено кафедрою РТС (протокол № 06/2024 від 27.06.2024)

Погоджено методичною комісією факультету/ННІ (протокол № 06/2024 від 28.06.2024)