

ПЕРЕДМОВА

Важливою складовою навчального процесу є лабораторні заняття, в ході яких студенти у спеціально обладнаних лабораторіях та під керівництвом викладача виконують експериментально-дослідницьку роботу в межах конкретної дисципліни з метою набуття практичних навичок за обраною ними спеціальністю.

Якщо під час лекцій, як одного з найважливіших видів навчальних занять, студентам надаються основи наукових знань, формується науковий світогляд, розкриваються найбільш складні питання матеріалу дисципліни “Модульне програмування”, то лабораторні заняття спрямовані на прищеплення практичних навичок, тобто реалізують один з головних принципів навчання – зв’язок Теорії з Практикою.

В забезпеченні означеного ключову роль відіграють організаційно-методична складова навчального процесу та відповідальне ставлення до нього студента. Відповідно до Положення про організацію освітнього процесу у Кіровоградському національному технічному університеті, лабораторне заняття включає проведення інструктажу з техніки безпеки, поточного **контролю підготовленості студента** до виконання конкретної лабораторної роботи, виконання завдань з теми заняття, оформлення індивідуального звіту з виконаної роботи та його захист перед науково-педагогічним працівником. Водночас, в межах самостійної роботи **у вільний від аудиторних занять час студент зобов’язаний**, зокрема, опрацьовувати навчальний матеріал дисципліни, літературні джерела та здійснювати належну підготовку до виконання лабораторних робіт.

Отже, слід **ретельно готуватись до кожного заняття**. Підготовка до чергової лабораторної роботи здійснюється студентом самостійно з обов’язковим опрацюванням навчальної, довідникової, наукової літератури задля ґрунтовного вивчення теоретичних положень дисципліни “Модульне програмування”, винесених на лабораторну роботу, а також самоконтролем підготовленості до виконання завдань за темою заняття.

САМОСТІЙНА ПІДГОТОВКА ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 6

Перелік теоретичних положень дисципліни

Для допуску до виконання лабораторної роботи № 6, мета якої полягає у підтвердженні під час лабораторного заняття теоретичних положень *лекцій 13-16 та CPC 4, 5* навчальної дисципліни “Модульне програмування”, студент **повинен знати** наступне:

- методи об’єктного аналізу й проектування;
- об’єктна методологія програмування;
- перевантаження функцій C/C++;
- параметри функцій за замовчуванням у C/C++;
- препроцесорні директиви, макроси й макрооператори C/C++;
- абстрактні типи даних (ADT);
- класи C++:
 - визначення (опис) класу як ADT;
 - члени класів;
 - спеціальні функції-члени класів;
 - доступ до членів класу;
 - об’єкти (екземпляри) класів;
- дозвіл області видимості в просторі імен C++;
- методика створення заголовкових файлів з ADT C++ в та їх використання під час реалізації програмних засобів.

СПИСОК РЕКОМЕНДОВАНОЇ ЛІТЕРАТУРИ

1. McConnell, Steve. Code Complete. Second Edition / Steve McConnell. – Redmond, Washington: Microsoft Press, 2010. – 960 p. (Макконнелл С. Совершенный код. Мастер-класс / Пер. с англ. / Стив Макконнелл – М. : Издательство «Русская редакция», 2010. – 896 с.).
2. Booch, Grady. Object-Oriented Analysis and Design with Applications (3rd Edition) / Grady Booch, Robert A. Maksimchuk,

- Michael W. Engle. – New York: Addison-Wesley Professional, 2007. – 720 p. (Буч Г. Объектно-ориентированный анализ и проектирование с примерами приложений [Текст] / Гради Буч и др. – 3-е изд. – М.: Вильямс, 2008. – 718 с.).
3. Довідник з препроцесора C/C++ [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Microsoft, 2016. – Режим доступу: <https://msdn.microsoft.com/uk-ua/library/y4skk93w.aspx> (дата звернення: 11.02.2016). – Назва з екрану.
 4. International Standard ISO/IEC 14882:2014 – Information technology – Programming languages – C++. – ISO/IEC, 2014.
 5. Липпман С. Язык программирования C++. Базовый курс, 5-е изд. : Пер. с англ. / Липпман Стенли, Лажоие Жози, Му Барбара. – М.: Вильямс, 2014. – 1120 с.
 6. Прата С. Язык программирования C++. Лекции и упражнения, 6-е изд. : Пер. с англ. / Стивен Прата. – М.: Вильямс, 2012. – 1248 с.
 7. Класи в C [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – CppStudio, 2016. – Режим доступу: <http://cppstudio.com/uk/post/439/> (дата звернення: 11.02.2016). – Назва з екрану.

ЗАПИТАННЯ І ЗАВДАННЯ ДЛЯ САМОКОНТРОЛЮ ПІДГОТОВЛЕНOSTІ ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 6

1. В чому полягає сутність процесу абстрагування як теоретичного метода пізнання на основі емпіричних даних, фактів тощо? Виходячи з означеного, дайте визначення поняттю “абстракція”.
2. Що визначають й описують під час об’єктної декомпозиції задачі?
3. Сформулюйте сутність і результати концептуалізації предметної області як метода пізнання (дослідження).
4. Що розуміють під аналізом сутності предметної області? Сформулюйте структуру або алгоритм об’єктного аналізу предметної області задачі.
5. Сформулюйте сутність методики створення заголовкового файлу з реалізованим класом – абстрактним типом даних.

6. Що є результатом реалізації процесу визначення інтерфейсу об’єкта предметної області та яким чином він описується мовою програмування C++?
7. В чому полягає мета перевантаження функцій під час реалізації проектних рішень мовою програмування C/C++?
8. Яким чином перевантаження функцій реалізовується мовою програмування C/C++? Наведіть приклад.
9. Перелічіть випадки (задачі), за яких є доцільним використання препроцесорних макросів `__FILE__`, `__LINE__`, `__DATE__`, `__TIME__`, `__TIMESTAMP__`.
10. Яким є синтаксис запису функції C/C++ з параметрами за замовчуванням?
11. Яке призначення препроцесорних директив `#define`, `#undef`, `#if`, `#ifdef`, `#ifndef`, `#error`, `#pragma`?
12. В чому полягає концепція абстрактних типів даних (ADT)? Чим вони відрізняється від вбудованих типів?
13. Здійсніть порівняльний аналіз типів даних мови C++ клас (`class`) та структура (`struct`).
14. Яким є синтаксис запису абстрактного типу даних клас (`class`) мовою програмування C++?
15. Яке призначення оператора `::` мови програмування C++ та синтаксис його запису? Наведіть приклад запису даних у глобальний об’єкт (змінну), ідентифікатор якого співпадає з ім’ям локального об’єкта (змінної).
16. Чим синтаксис запису даних-членів класу (`class`) відрізняється від опису членів структури (`struct`) C++?
17. Якими є синтаксис і правила запису функцій-членів класу C++?
18. Які розрізняють рівні (секції) доступу до членів класу C++ та за допомогою яких специфікаторів вони визначаються?
19. Яким є синтаксис оголошення (створення) об’єкта класу C++? Здійсніть порівняльний аналіз класу і об’єкта цього класу.
20. Яке призначення та синтаксис запису спеціальних функцій-членів класу C++ (конструктори, деструктори)?