

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 самоконтролю підготовленості до виконання завдань за темою заняття.

САМОСТІЙНА ПІДГОТОВКА ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 5

Перелік теоретичних положень дисципліни

Для допуску до виконання лабораторної роботи № 5, мета якої полягає у підтвердженні під час лабораторного заняття теоретичних положень *лекцій 8, 11, 12 та СРС 4* навчальної дисципліни “Модульне програмування”, студент **повинен знати** наступне:

- типи даних C/C++:
 - структури;
 - вказівники;
 - посилання;
- керування динамічною пам’яттю засобами C/C++;
- динамічні структури даних:
 - лінійні списки;
 - стеки;
 - черги;
 - бінарні дерева;
- методи оброблення динамічних структур даних;
- потокові класи C++;
- бінарні потоки (файли).

Список рекомендованої літератури

1. International Standard ISO/IEC 14882:2014 – Information technology – Programming languages – C++. – ISO/IEC, 2014.
2. Павловская Т. А. C/C++. Структурное и объектно-ориентированное программирование: Практикум / Т. А. Павловская, Ю. А. Щупак. – СПб.: Питер, 2011. – 352 с.
3. Прата С. Язык программирования C++. Лекции и упражнения, 6-е изд. : Пер. с англ. / Стивен Прата. – М.: Вильямс, 2012. – 1248 с.

4. Мова програмування C++ [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – CppStudio, 2016. – Режим доступу: <http://cppstudio.com/uk/cat/274/> (дата звернення: 15.01.2016). – Назва з екрану.
5. Стандартні бібліотеки та мова C++ [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – Microsoft, 2016. – Режим доступу: <https://msdn.microsoft.com/uk-ua/library/hh875057.aspx> (дата звернення: 15.01.2016). – Назва з екрану.
6. Липпман С. Язык программирования C++. Базовый курс, 5-е изд. : Пер. с англ. / Липпман Стенли, Лажоие Жози, Му Барбара. – М.: Вильямс, 2014. – 1120 с.
7. Програмування по-українськи : C++ [Електронний ресурс] : [Веб-сайт]. – Електронні дані. – programming.in.ua, 2015. – Режим доступу: <http://programming.in.ua/programming/c-plus-plus.html> (дата звернення: 27.01.2016). – Назва з екрану.

ЗАПИТАННЯ І ЗАВДАННЯ ДЛЯ САМОКОНТРОЛЮ ПІДГОТОВЛЕНOSTІ ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 5

1. Що у програмуванні розуміють під часом життя динамічних змінних (об'єктів)?
2. Дайте визначення вказівникові та посиланню. Перелічіть випадки, у яких їх варто й доцільно використовувати.
3. Яким є синтаксис оголошення вказівників у C/C++? Наведіть приклад оголошення вказівника та його ініціалізації адресою існуючого об'єкта (змінної).
4. Скільки пам'яті виділяється для порожнього вказівника? Відповідь обґрунтуйте і доведіть експериментально.
5. Перелічіть допустимі операції над вказівниками та оператори C/C++, які їх реалізують.
6. Яке призначення ключових слів `new` та `delete` у C/C++?
7. В чому полягає відмінність між виконанням операції `delete` та `delete[]` у C/C++?

8. Що міститиме `stm` при: `short ptt,*stm = &ptt; ?`
9. Яке ключове слово мови програмування C/C++ реалізує операцію визначення розміру змінної (об'єкта) або типу даних?
10. Яким чином у C/C++ реалізовується опосередкована адресація?
11. Мовою програмування C/C++ наведіть приклад створення динамічного одновимірної масива та звільнення виділеної для нього пам'яті.
12. Що буде включено у стандартний потік при:
`double *dptr;`
`short *sptr;`
`cout << (sizeof dptr - sizeof sptr);`
Відповідь обґрунтовано поясніть.
13. Яка константа класу `ios` (бітова маска 80h) C++ призначена для відкриття файлового потоку в двійковому режимі? Наведіть приклад відкриття бінарного файла для запису даних у його кінець.
14. Яка функція-член об'єкта `ofstream` забезпечує читання заданої кількості байт з асоційованого потоку в змінну-буфер, а яка – включення (запис) даних у потік.
15. Що буде записано у файл, якщо в асоційований з ним бінарний потік включити рядок символів?
16. Дайте визначення наступним динамічним структурам даних: список, черга, стек, бінарне дерево. Перелічіть випадки (задачі), у яких є доцільним використання кожної з них.
17. В чому полягає відмінність одно- від двонаправленого списку з погляду їх реалізації? Наведіть приклад опису мовою C/C++ елемента одно- та двонаправленого списків і виділення для них динамічної пам'яті.
18. Перелічіть допустимі операції над лінійними списками як простими динамічними структурами даних.
19. Які розрізняють потоки і в чому полягає їх відмінність з погляду використання під час реалізації ПЗ?
20. Сформулюйте алгоритм запису в двійковий (бінарний) файл однонаправленого списку. Чим він відрізняється від алгоритмів зберігання у файл стека та черги?