

**МЕТОДИЧНІ ВКАЗІВКИ
ДО САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ
при вивченні дисципліни
«ТЕХНОЛОГІЯ ПРОГРАМУВАННЯ»**

для студентів галузі знань 12 «Інформаційні технології»
спеціальності 125 «Кібербезпека»
(«Безпека інформаційних і комунікаційних систем»)

Міністерство освіти і науки України
Вінницький національний технічний університет
Факультет інформаційних технологій та комп'ютерної інженерії
Кафедра захисту інформації

**МЕТОДИЧНІ ВКАЗІВКИ
ДО САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ
при вивченні дисципліни
«ТЕХНОЛОГІЯ ПРОГРАМУВАННЯ»**

для студентів галузі знань 12 «Інформаційні технології»
спеціальності 125 «Кібербезпека»
(«Безпека інформаційних і комунікаційних систем»)

Вінниця
ВНТУ
2018

Рекомендовано до друку Методичною радою Вінницького національного технічного університету Міністерства освіти і науки України (Протокол № від 2018 р.)

Рецензенти:

Н. Р. Кондратенко, кандидат технічних наук, професор

Д. І. Кательніков, кандидат технічних наук, доцент

Технологія програмування: Методичні вказівки до самостійної роботи студентів / уклад. В. А. Каплун, А. В. Остапенко-Баженова, Лукічов В. В. – Вінниця: ВНТУ, 2018. – 36 с.

У даних методичних вказівках наводяться основні рекомендації щодо організації самостійної роботи студентів при вивченні теоретичного матеріалу та підготовці до лабораторних занять з дисципліни «Технологія програмування». Наведено основні питання змістових модулів, ключові моменти, на які потрібно звернути увагу при засвоєнні кожної теми. Крім того, по кожній темі наведено контрольні питання.

Методичні вказівки призначені для студентів денної і заочної форм навчання із спеціальності 125 «Кібербезпека» (спеціалізації «Безпека інформаційних і комунікаційних систем») при виконанні лабораторних робіт з дисципліни «Технологія програмування».

ВСТУП

Метою викладання дисципліни "Технологія програмування" є формування знань, умінь та навичок для проектування комп'ютерних програм з використанням сучасних технологій структурного, процедурного та об'єктно-орієнтованого програмування, набуття знань та навичок для реалізації основних алгоритмів обробки інформації різного типу.

Під час вивчення дисципліни студентам надається систематизована інформація про технології програмування засобами процедурної та об'єктно-орієнтованої мов програмування C та C++, основи програмування для операційних систем сімейства Windows та способи розробки графічного інтерфейсу повноцінних додатків.

Основними завданнями при вивченні дисципліни "Технологія програмування" є засвоєння сучасних і перспективних методів та технологій побудови програмних продуктів, засвоєння засобів і методів для здійснення тестування, верифікації, налагоджування програм, використовуючи для цього інтегровані програмні середовища.

Окрім основних задач з набуття теоретичних знань і практичних навичок з дисципліни "Технологія програмування" велика увага приділяється засвоєнню основних правил здійснення алгоритмізації задач, розробці різних видів схем, створенню програмної документації для супроводження розроблених програмних засобів.

1 СТРУКТУРНИЙ ПІДХІД



1.1 Основні підходи до програмування

➤ Парадигми програмування [1]:

- процедурна - необхідно визначити потрібні процедури і використати найкращі алгоритми;
- об'єктна - необхідно вирішити, які модулі потрібні, а потім розбити їх так, щоб приховати дані в цих модулях;
- об'єктно-орієнтована – необхідно визначити, які класи потрібні, описати повний набір операцій для кожного класу і виразити спільність через успадкування;
- узагальнена – необхідно визначити, які алгоритми потрібні, і параметризувати їх так, щоб вони могли працювати із більшістю потрібних типів і структур даних.

➤ Історія розвитку мов програмування складається з ряду етапів [2, 3]:

- 40-50 роки ХХ ст. – машинні мови;
- початок 50-х років ХХ ст. – Асемблер;
- середина 50-х років ХХ ст. – перші мови програмування високого рівня (Fortran, Cobol, Basic, Code);
- 80-і роки ХХ ст. – алгоритмічні мови програмування (Pascal, Сі);
- 90-і роки ХХ ст. – мови об'єктно-орієнтованого програмування (С++, Object Pascal, Delphi, Visual Basic, Qbasic);
- 90-і роки ХХ ст. – мови програмування для комп'ютерних мереж (Java, JavaScript);
- початок ХХІ ст. – мови програмування на платформі .NET (Visual Basic.Net, Visual C#, Visual J#).

Класифікація мов програмування може бути здійснена за різними ознаками [1, 3]:

- за способом трансляції: компільовані та інтерпретовані;
- за рівнем (особливостям побудови): машино-орієнтовані (асемблери) та універсальні (процедурно-орієнтовані і об'єктно-орієнтовані).

Основні етапи проектування комп'ютерної програми [2, 3]:

- постановка завдання;
- формалізація (математична постановка задачі);
- вибір (або розроблення) методу розв'язування;
- розроблення алгоритму;
- складання програми;
- налагодження програми;
- обчислення та обробка результатів.
- документування та ін.

Контрольні запитання

1. Охарактеризуйте основні парадигми програмування?
2. Які основні етапи розвитку мов програмування?
3. До якого етапу відносять мови програмування С та С++?
4. Які мови програмування на платформі .NET ви знаєте?
5. Як класифікувати мови програмування за способом трансляції?
6. На які групи розділяють мови програмування за особливостями будови?
7. Що таке комп'ютерна програма?
8. Наведіть основні етапи проектування комп'ютерної програми і охарактеризуйте кожний з них.
9. В чому роль етапу формалізації при розробці програми?
10. Поясніть поняття компілятора, транслятора та інтерпретатора.
11. В чому полягає роль компонувача?
12. Яка різниця між статичними, об'єктними бібліотеками та бібліотеками динамічного компонування?



Список джерел

1. Курносів О. П. Інформатика / Кулев С. А., Улезько А. В. та ін. Під ред. О. П. Курносова. – М. : Колосс, 2005. – 299 с.
2. Островський В. А. Інформатика: учеб. для вузів. М. : Вища школа, 2000. – 511 с.
3. Семакин І. Інформатика: Базовий курс / Семакин І., Залогова Л., Русаков З., Шестакова Л. – М. : БИНОМ, 2005. – 105 с.



1.2 Теорія алгоритмів. Структурування алгоритмів

Алгоритм – система формальних правил, що визначає зміст і порядок дій над вхідними даними і проміжними результатами, необхідними для отримання кінцевого результату при розв'язуванні задачі [4].

Властивості алгоритмів [5]:

- дискретність полягає в тому, що виконання чергової операції починається у певний момент часу, моменти часу не обов'язково еквідистантні;
- детермінованість означає, що перед виконанням першої операції відомі початкові дані, у будь-який момент часу точно відомі наступні операції (прогнозованість) та результат попередньої операції;
- впорядкованість – встановлена черговість операції (інакше — відношення порядку на їх множині);
- масовість означає, що алгоритм розв'язує задачу, що належить до певного класу задач;

- елементарність означає, що завжди можна вказати елементарну (виконувану апаратно, не програмовану) операцію.

Псевдокод – система позначень і правил, призначена для записування алгоритмів. Він займає проміжне місце між звичайною і формальною мовою. Через свої особливості псевдокоди орієнтовані на людину. У псевдокоді не вимагається дотримання синтаксичних правил для запису команд, які властиві формальним мовам, що полегшує запис алгоритму на стадії його проектування і дає можливість використати ширший набір команд, розрахований на абстрактного виконавця [6].

Блок-схеми. При використанні цієї форми подання алгоритму процес розв'язання поділяється на окремі етапи (або операції), які зображуються у вигляді спеціальних блоків, конфігурація яких вказує тип дій. Стрілки, що з'єднують, задають послідовність виконання цих дій. Конфігурація та розміщення блоків визначені відповідними стандартами ЄСПД (Єдиної системи програмної документації) [6, 7].

Основні структури алгоритмів – це обмежений набір блоків і стандартних способів їх з'єднання для виконання типових послідовностей дій. Використання декількох основних структур дає можливість будувати різноманітні алгоритми. До основних структур алгоритмів належать:

- лінійна або послідовна без будь-яких розгалужень конфігурація алгоритму, що нагадує форму ланцюжка;
- розгалужена конфігурація алгоритму, що містить в собі як послідовності, так і розпаралелення послідовностей. Використовується, коли залежно від умови потрібно виконати ту чи іншу дію, здійснити обхід, якщо одна вітка не містить жодних дій, здійснити множинний вибір, коли умова має більш як три можливі варіанти;
- циклічна, щовикористовується при необхідності виконувати деякі дії декілька разів.

Основні види схем. Схема в програмній документації – це графічне представлення визначення, аналізу або методу розв'язування задачі, в якому використано символи для відображення операцій, даних, потоку, обладнання тощо.

При розробці програм використовують такі схеми:

- схеми даних відображають шлях даних при розв'язанні задач і визначають етапи обробки, а також різні використовувані носії даних;
- схеми програм відображають послідовність операцій в програмі;
- схеми роботи системи відображають управління операціями і потік даних в системі;
- схема взаємодії програм відображає шлях активацій програм і взаємодій з відповідними даними. Кожна програма в схемі взаємодії програм показується тільки один раз (у схемі роботи

системи програма може відобразитися більше, ніж в одному потоці управління).;

- схема ресурсів системи відображає конфігурацію блоків даних, блоків обробки цих даних, яка потрібна для розв'язання задачі або набору задач.

Формування алгоритмів на основі низхідного покрокового уточнення.

Основи алгоритмізації.

Правила виконання схем алгоритмів.

Критерій ефективності алгоритмів.

Алгоритмічна культура – це частина загальної математичної культури і загальної культури мислення, яка зумовлює формування вмінь, пов'язаних з

розумінням суті поняття алгоритму і його властивостей, тобто це сукупність

знань, умінь і навичок, які дозволяють успішно розв'язувати задачі.

У процесі розв'язування прикладних задач вибір алгоритму має певні труднощі. Одна з основних проблем, що виникають при переході від алгори-

тмів безкомп'ютерних до алгоритмів, представлених у вигляді комп'ютерної

програми, пов'язана з формою їх запису, яка повинна бути зрозумілою комп'ютеру. Алгоритм повинен відповідати наступним вимогам:

- 1) бути простим для розуміння, переведення в програмний код та налагодження;

1.3 Мови програмування

Вступ у програмування мовою C. Елементи мови C: алфавіт, ідентифікатори, константи, коментарі. Структура програми на C. Базові типи даних. Перетворення типів. Функції введення та виведення. Основні операції: арифметичні, присвоювання, порівняння, логічні, побітові, операція слідування умовна операція та операція визначення розміру даних. Ключові слова мови C. Пріоритет операцій.

1.4 Структурна розробка програм на C

Структури вибору: if, if/else, switch. Структури повторення: while, for, do/while. Оператор розриву break. Оператор продовження continue. Оператор переходу goto.

1.5 Функції в C

Функції математичної бібліотеки. Визначення функцій. Прототипи функцій. Виклик функцій. Генерація випадкових чисел. Рекурсивні функції. Класи пам'яті. Рекурсія в порівнянні з ітерацією.

1.6 Арифметика покажчиків у мовах програмування

Вказівники в C. Оголошення та ініціалізація змінної-вказівника. Операції над вказівниками. Передача параметрів по посиланню. Масиви вказівників. Вказівники на функції.

1.7 Динамічне виділення пам'яті

Порівняння динамічного і статичного розміщення даних у пам'яті. Поняття купи. Покажчики на змінні різних типів. Арифметика покажчиків.. Використання модифікатора `const` з вказівниками.

1.8 Робота з масивами у C та у C++

Масиви в C. Оголошення масивів. Одновимірні і багатовимірні масиви. Статичне і динамічне оголошення масивів. Передача масивів функціям.

1.9 Символи та символні рядки рядки в C та C++

Основні відомості про символи та рядки. Функції бібліотеки обробки символів. Функції перетворення рядків. Функції стандартної бібліотеки введення/виведення. Функції бібліотеки обробки рядків: операції над рядками, порівняння, пошук. Функції пам'яті з бібліотеки обробки рядків.

1.10 Клас `string` і потокова обробка рядків

Клас `String`. Поняття потоків символів. Присвоєння і конкатенація рядків. Порівняння рядків. Підрядки. Перестановка рядків. `String`-характеристики. Пошук символів в рядку. Заміна символів в рядку. Вставка символів в рядок. Ітератори. Потокова обробка рядків.

1.11 Структури та об'єднання в C і в C++

Основні питання

Опис структур. Ініціалізація структур. Доступ до елементів структур. Використання структур з функціями. Об'єднання. Бітові поля. Перераховані константи. Масиви структур.

1.12 Робота з файлами в С і С++

Основні питання

Текстові файли з послідовним доступом. Текстові файли з довільним доступом. Двійкові файли. Використання дескрипторів файлів. Буферизація потоків. Файли послідовного доступу. Файли довільного доступу. Введення/виведення об'єктів.

1.13 Основні методи сортування масивів

Поняття сортування. Підходи до сортування. Сортування масивів: метод бульбашки, метод вибору, сортування вставками. Пошук в масивах: лінійний та бінарний. Багатовимірні масиви. Сортування методом Шелла, блокове сортування. Швидке сортування.

1.14 Динамічні структури даних в С

Поняття динамічних структур. Застосування динамічних структур в операційних системах та у базах даних. Структури, що посилаються на себе. Зв'язані списки. Стеки. Черги. Деревя.

ОБ'ЄКТНО-ОРІЄНТОВАНИЙ ПІДХІД

Тема 15. Інтегровані середовища програмування.

Сучасні інтегровані програмні середовища. Редактор сирцевого коду, компілятор, інтерпретатор, засоби автоматизації збірки, системи контролю версій, засоби для профілювання, засоби та утиліти для спрощення розробки графічного інтерфейсу користувача, оглядач класів, інспектор об'єктів та діаграм ієрархії класів для використання ООП.

Тема 16. Тестування і верифікація програм.

Верифікація і валідація як методи перевірки правильності виконання заданих функцій. Процес верифікації. Способи верифікації програм та її результати. Основні концепції тестування (Testing Basic Concepts and definitions), рівні тестування (Test Levels), техніки тестування (Test Techniques), метрики тестування (Test Related Measures), керування процесом тестування (Managing the Test Process).

Тема 17. Компонентне програмування та засоби налагодження програм

Компонентне програмування як подальший розвиток ООП, повторне використання, специфікації компонентів і їхніх інтерфейсів, композиції та конфігурації компонентів. Зв'язки між компонентами та об'єктні зв'язки. Основні методи налагодження програми. Захисне програмування. Методологічні принципи налагодження програм. Автономне та комплексне налагодження програмного засобу.

Тема 18. Основи об'єктно-орієнтованої технології програмування.

Загальні поняття об'єктно-орієнтованого програмування (ООП).

Основні концепції і принципи ООП. Абстракція даних. Механізм інкапсуляції. Успадкування. Ієрархія класів. Процес поліморфізму.

Тема 19. Проблема опису довільних типів даних.

Стандартна бібліотека шаблонів (STL). Вступ в STL. Контейнери послідовностей. Асоціативні контейнери. Адаптери контейнерів. Алгоритми. Класи C++ та абстрагування даних. Область дії класу та доступ до елементів класу. Відділення інтерфейсу від реалізації. Керування доступом до елементів класу. Конструктори. Деструктори. Використання елементів-даних та елементів-функцій. Дружні функції та дружні класи. Показчик this. Статичні елементи класу. Абстракція даних і приховування інформації.

Тема 20. Поліморфізм. Перевантаження операцій в C++.

Основи перевантаження операцій. Обмеження на перевантаження операцій. Функції-операції як члени класу та як дружні функції. Перевантаження операцій «помістити в потік» та «взяти з потоку». Перевантаження одномісних та двомісних операцій. Перевантаження ++ та --. Перетворення типів. Віртуальні функції C++ та поліморфізм. Віртуальні функції. Абстрактні базові класи та конкретні класи.

Поліморфізм. Нові класи та динамічне зв'язування. Віртуальні деструктори.

Тема 21. Успадкування в C++.

Успадкування: базові та похідні класи. Захищені члени класу. Приведення типів покажчиків базових класів до покажчиків похідних класів. Використання функцій-членів. Перевизначення членів базового класу в похідному класі. Використання конструкторів і деструкторів в похідних класах. Неявне перетворення об'єктів похідних класів в об'єкти базових класів. Множинне успадкування.

Семестр 3-й

Змістовий модуль 2. Об'єктно-орієнтований підхід (продовження)

Тема 22. Прикладний програмний інтерфейс Windows.

Особливості програмування під Windows. Обробка повідомлень Windows. Послідовність виконання обробки повідомлень в операційних системах. Основні групи повідомлень. Процедури обробки повідомлень вікон. Особливості використання інформаційних вікон повідомлення і їх види.

Тема 23. Концепція типів даних.

Шаблони C++. Шаблони класів. Шаблони класів і нетипові параметри. Шаблони та успадкування. Шаблони та друзі. Шаблони та статичні члени. Основні типи даних для програмування під Windows. Основні класи для опису типів даних. Угорська нотація. Вікна – основні об'єкти Windows. Дескриптор вікна. Хендл вікна. Види вікон.

Тема 24. Розробка додатків в інтегрованих середовищах програмування.

Основи програмування у WINDOWS. Структура головної функції додатку. Відмінності при написанні додатків MFC і API. Переваги і недоліки використання API-функцій. Обробка виключних ситуацій в C++. Основи обробки винятків. Генерація винятків. Перехоплення винятків. Повторна генерація винятків. Специфікації винятків. Обробка неочікуваних винятків. «Розкручування» стеку.

Тема 25. Розробка графічного інтерфейсу у додатках.

Виведення інформації у вікна Windows. Контекст пристрою. Методи виведення інформації у вікна. Виведення текстової інформації. Робота зі шрифтами, вирівнюванням тексту. Створення власних шрифтів. Логічні об'єкти графіки. Кольори, пензлі, пера. Виведення у вікно графічних примітивів. Виведення у вікно растрових зображень. Поняття подвійної буферизації. Способи усунення мерехтіння.

Тема 26. Елементи керування і діалогові вікна. Обробка повідомлень.

Діалогові вікна. Процедура обробки діалогового вікна. Обробка повідомлень діалогового вікна. Основні елементи керування в діалогових вікнах. Стандартні діалогові вікна для вибору кольору, шрифтів, панель зчитування та зберігання файлів.

Тема 27. Додаткові засоби для оформлення додатків Windows.

Методи обробки переривань від таймера. Створення динамічних зображень. Імітація руху у програмах. Створення анімації. Створення панелі інструментів та вікон підказок, підказок меню, статусних вікон тощо, таблиці акселераторів. Динамічне і плаваюче меню.

Змістовий модуль 3. Розподілені системи.

Тема 28. Системне програмне забезпечення

Системні програмні засоби, класифікація системного ПЗ. СПЗ для забезпечення діяльності прикладних програмних систем. Операційні системи. Тестові та діагностичні програми. Антивірусні програми. Командно-файлові процесори (оболонки).

Тема 29. Статичне планування і динамічне планування

Розподілені комп'ютерні системи. Дисципліни планування в них. Особливості динамічного планування. Особливості динамічного планування в розподілених системах. Позитивні характеристики динамічного планування.

Рекомендована література

Базова

- 1. Курносів О. П. Информатика / Кулев С. А., Улезько А. В. та ін. Під ред. О. П. Курносова. – М. : Колосс, 2005. – 299 с.**
- 2. Островський В. А. Информатика: учеб. для вузів. М. : Вища школа, 2000. – 511 с.**
- 3. Семакин І. Информатика: Базовий курс / Семакин І., Залогова Л., Русаков З., Шестакова Л. – М. : БИНОМ, 2005. – 105 с.**
- 4. Яворський Б. І. Теорія алгоритмів. Конспект лекцій. – Тернопіль: ТДТУ імені Івана Пулюя, 2000. – 36 с.**
- 5. Єжова Л. Ф. Алгоритмізація і програмування процедур обробки інформації: Навч.-метод. посібник для самост. вивч. дисц. / Л. Ф. Єжова. – К. : КНЕУ, 2000. – 152 с.**
- 6. Васильків Н. М., Опорний конспект лекцій з дисципліни “Основи алгоритмізації” спеціальність “Комп’ютерні системи та мережі”, освітньо-кваліфікаційний рівень “Бакалавр” – Тернопіль : Економічна думка, 2005. – 32 с.**
- 7. ЄСПД**

8. Дейтел Х. М. Как программировать на С. / Х. М. Дейтел, П. Дж. Дейтел [3-е издание]. – М. : Бином-Пресс, 2002. – 1168 с.
9. Глушаков С. В. Язык программирования С++ / Глушаков С. В., Коваль А. В., Смирнов С. В. – Харьков: Фолио, 2001. – 500 с. – ISBN 966-03-1282-2.
10. Дейтел Х. М. Как программировать на С++: 5-е издание / Х. М. Дейтел, П. Дж. Дейтел ; пер. с англ. В. В. Тимофеев. – М: Бином-Пресс, 2008. – 1456 с. – ISBN 978-5-9518-0224-8 (в пер.).
11. Программирование алгоритмов защиты информации. Учебное пособие / [Домашев А. В., Попов В. О., Правиков Д. И. и др.] – М.: Нолидж, 2002. – 416 с. – ISBN: 5-89251-102-2.
12. Кнут Д. Е. Искусство программирования, том 1. Основные алгоритмы, 3-е изд. / Д. Е. Кнут. – М.: «Вильямс», 2001. – 720 с. – ISBN 5-8459-0080-8 (в пер.).
13. Кнут Д. Е. Искусство программирования, том 3. Сортировка и поиск, 2-е изд. / Д. Е. Кнут. – М.: «Вильямс», 2001. – 832 с. – ISBN 5-8459-0082-4 (в пер.).
14. Круподьорова Л. М. Технологія програмування мовою Сі. Частина 1. Навчальний посібник / Л. М. Круподьорова, А. М. Петух. – Вінниця: ВНТУ, 2006. – 206 с.
15. Технологія програмування. Лабораторний практикум : навчальний посібник / В. А. Каплун, Ю. В. Баришев, А. В. Остапенко – Вінниця: ВНТУ, 2015. – 125 с.

16. **Круподьорова Л. М. Технологія програмування мовою Сі. Частина 2. Лабораторний практикум / Л. М. Круподьорова, А. М. Пстух. – Навчальний посібник. – Вінниця: ВНТУ, 2006. – 184 с.**
17. **Подбельский В. В. Программирование на языке Си / В. В. Подбельский, С. С. Фомин. – М: Финансы и статистика, 2004. – 600 с. – ISBN 5-279-02180-6.**
18. Семеренко В. П. Програмування мовами С та С++ в середовищі Windows. Навчальний посібник / В. П. Семеренко. – Вінниця: ВДТУ, 2002. – 128 с.
19. **Хаймен М. Borland С++ для "чайников" / М. Хаймен. – К: Диалектика, 1995. – 416 с. – ISBN 5-7707-7921-7 (в пер.).**
20. **Гудман С., Хидетниemi С. Введение в разработку и анализ алгоритмов / С. Гудман, С. Хидетниemi. – М.: Мир, 1981 – 368 с.**
21. **Искусство программирования на С. Фундаментальные алгоритмы, структуры данных и примеры приложений. Энциклопедия программиста / [Хэзфилд Р., Кирби Л. и др.] – М: Диасофт, 2001. – 736 с. – ISBN 966-7393-82-8 (в пер.).**
22. **Шилдт Г., Полный справочник по С / Г. Шилдт. – СПб.: Вильямс, 2003. – 800 с. – ISBN 5-8459-0226-6 (в пер.).**
23. Ганеев Р. М. Проектирование интерфейса пользователя средствами Win32 API / Р. М. Ганеев. – М.: Горячая линия - Телеком, 2001. – 334 с. – ISBN 5-93517-050-7.
24. Румянцев П. В. Азбука программирования в Win32 API / П. В. Румянцев. – М.: Горячая линия - Телеком, 2001. – 310 с. – ISBN 5-93517-048-5.
25. Щупак Ю. А. Win32 API. Эффективная разработка приложений / Ю. А. Щупак. – СПб.: Питер, 2007. – 572 с. – ISBN 5-469-01361-8.
26. **Глушаков С. В. Программирование на Visual С++ 6.0 / Глушаков С. В., Коваль А. В., Черепнин С. А. – Харьков: Фолио, 2002. – 726 с. – ISBN 966-03-1776-X.**
27. **Методичні вказівки до виконання курсової роботи з дисципліни "Програмування" для студентів напряму підготовки 6.170101 "Безпека інформаційних і комунікаційних систем" /**

Уклад. В. А. Каплун, О. П. Войтович. – Вінниця: ВНТУ, 2010. – 62 с.

Додаткова

1. Алгоритмы: построение и анализ, 2-е издание / Т. Х. Кормен, Ч. И. Лейзерсон, Р. Л. Ривест, К. Штайн ; пер. с англ. И. В. Красикова и др. – М. : Издательский дом "Вильямс", 2005. – 1296 с. – ISBN 5-8459-0857-4 (в пер.).
2. Лаптев В. В. С++. Объектно-ориентированное программирование: Учебное пособие / В. В. Лаптев. – СПб.: Питер, 2008. – 464 с. – ISBN 978-5-91180-200-4.
3. Керниган Б. У. Язык программирования С, 2-е издание / Б. У. Керниган, Д. М. Ритчи ; пер. с англ. В. Л. Бродового. – М. : Издательский дом "Вильямс", 2009. – 304 с. – ISBN 978-5-8459-0891-9 (в пер.).
4. Проценко В. С. Техніка програмування мовою СІ / Проценко В.С., Чаленко П.Й., Ставровський А.Б. – К.: Либідь, 1993. – 231 с.
5. Страуструп Б. Язык программирования С++. Специальное издание / Б. Страуструп ; пер. с англ. Н. Н. Мартынова. – М.: Бином, 2011. – 1136 с. – ISBN 978-57989-0425-9 (в пер.).
6. Уэйт М. Язык Си. Руководство для начинающих / Уэйт М., Прата С. Мартин Д. – М.: Мир, 1988. – 512 с. – ISBN 5-03-001309-1 (в пер.).
7. Мешков А. А. Visual С++ и MFC. Том 1 / А. А. Мешков, Ю. Тихомиров. – СПб.: Питер, 2007. – 572 с. – ISBN 5-8206-0073-8.
8. Мешков А. А. Visual С++ и MFC. Том 2 / А. А. Мешков, Ю. Тихомиров. – СПб.: Питер, 2008. – 572 с. – ISBN 5-8206-0073-8.
9. Румянцев П. В. Работа с файлами в Win32 API / П. В. Румянцев. – М.: Горячая линия - Телеком, 2002. – 216 с. – ISBN 5-93517-097-3.
10. Интернет-ресурси.
11. ГОСТ 19.105-78. Единая система программной документации. Общие требования к программным документам.

Навчальне видання

**Валентина Аполінаріївна Каплун
Аліна Василівна Остапенко-Боженова
Віталій Володимирович Лукічов**

**МЕТОДИЧНІ ВКАЗІВКИ
ДО САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ
при вивченні дисципліни
"ТЕХНОЛОГІЯ ПРОГРАМУВАННЯ"
для студентів галузі знань 12 «Інформаційні технології»
спеціальності 125 «Кібербезпека»**

Редактор В. Дружиніна

Укладачі: Каплун Валентина Аполінаріївна
Остапенко-Боженова Аліна Василівна

Оригінал-макет підготовлено В. Каплун

Підписано до друку
Формат 29,7×42¼. Папір офсетний.
Гарнітура Times New Roman.
Друк різнографічний. Ум. друк. арк.
Наклад ... пр. Зам. № 2016-

Вінницький національний технічний університет,
навчально-методичний відділ ВНТУ,
21021, м. Вінниця, Хмельницьке шосе, 95,
ВНТУ, к. 2201.
Тел. (0432) 59-87-36.
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.

Віддруковано у Вінницькому національному технічному університеті
в комп'ютерному інформаційно-видавничому центрі.
21021, м. Вінниця, Хмельницьке шосе, 95,
ВНТУ, ГНК, к. 114.
Тел. (0432) 59-85-32,
publish.vntu.edu.ua; email: kivc.vntu@gmail.com.
Свідоцтво суб'єкта видавничої справи
серія ДК № 3516 від 01.07.2009 р.