

МІЖРЕГІОНАЛЬНА
АКАДЕМІЯ УПРАВЛІННЯ ПЕРСОНАЛОМ



МАУП

**МЕТОДИЧНІ МАТЕРІАЛИ
ЩОДО ЗАБЕЗПЕЧЕННЯ САМОСТІЙНОЇ
РОБОТИ СТУДЕНТІВ
з дисципліни
“СУЧАСНІ КОМП’ЮТЕРНІ ТЕХНОЛОГІЇ
ОБ’ЄКТНО-ОРІЄНТОВАНОГО
ПРОГРАМУВАННЯ НА C++”
(для бакалаврів)**

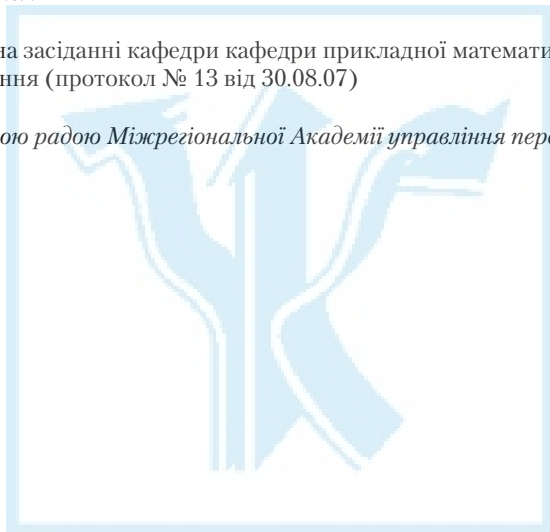
МАУП

Київ 2008

Підготовлено доцентом кафедри прикладної математики та програмування
В. Б. Зваридчуком

Затверджено на засіданні кафедри прикладної математики та програмування (протокол № 13 від 30.08.07)

Схвалено Вченою радою Міжрегіональної Академії управління персоналом



Зваридчук В. Б. Методичні матеріали щодо забезпечення самостійної роботи студентів з дисциплін “Сучасні комп’ютерні технології об’єктно-орієнтованого програмування на C++” (для бакалаврів). — К.: МАУП, 2008. — 36 с.

Методичні матеріали містять пояснювальну записку, тематичний зміст, питання щодо самостійного вивчення студентами, теми рефератів, тестові завдання, питання для самоконтролю та список літератури.

Призначена для методичного забезпечення самостійної роботи студентів, які вивчають навчальну дисципліну “Сучасні комп’ютерні технології об’єктно-орієнтованого програмування на C++”.

ПОЯСНЮВАЛЬНА ЗАПИСКА

Основний зміст самостійної роботи студентів над дисципліною полягає у вивченні та застосуванні системи знань у галузі теорії та практики сучасних комп'ютерних технологій об'єктно-орієнтованого програмування на C++, які застосовуються при виконанні лабораторних робіт.

До самостійної роботи належить також вивчення та освоєння методичних вказівок до лабораторних робіт і вивчення додаткової літератури, пов'язаної з виконанням цих робіт.

Лише постійне самостійне навчання дає можливість якомога повніше оволодіти такою сумою знань і вмінь, які дали б змогу заявити про себе як про професіонала. Студент, який хоче якомога краще оволодіти професією, має добре розуміти: на занятті викладач подає основи знань, вчить навчатися, виокремлює ті ключові положення дисципліни, які пробуджують у молодій людині потяг до поглиблення й удосконалення знань. Збагачення загальною сумою знань, накопичення людством, розширення загального світогляду, усвідомлення наявної перспективи щодо реалізації певних знань є основним мотивом до сумлінного ставлення до навчання. Самостійна навчальна діяльність студента буде результативною лише тоді, коли вона ґрунтуватиметься на внутрішній потребі. Виховання відповідної здатності у студента потребує чіткого узгодження процесу самоосвіти з цілями навчання та виховання.

Згідно з державними стандартами навчальний матеріал навчальної дисципліни, передбачений робочим навчальним планом для засвоєння студентом у процесі самостійної роботи, виноситься на підсумковий контроль поряд з навчальним матеріалом, який опрацьовувався при проведенні навчальних занять. Самостійна робота студента над засвоєнням навчального матеріалу з конкретної дисципліни може виконуватися у бібліотеці вищого закладу освіти, навчальних кабінетах, комп'ютерних класах (лабораторіях), а також у домашніх умовах. Самостійна робота студента повинна бути спланована, організаційно і методично спрямована як особиста творча праця без прямої взаємодії з викладачем. Час, відведений для самостійної роботи, регламентується робочим навчальним планом і згідно з Болонською декларацією повинен становити не менш 50 % загального обсягу навчального часу студента, відведеного для вивчення конкретної дисципліни. За необхідності ця робота проводиться від-

повідно до заздалегідь складеного графіка, що гарантує можливість індивідуального доступу студента до потрібних дидактичних засобів. Графік доводиться до відома студентів на початку поточного семестру. При організації самостійної роботи студентів із використанням складного обладнання, складних систем доступу до інформації (наприклад, комп'ютерних баз даних, систем автоматизованого проектування тощо) передбачається можливість отримання необхідної консультації або допомоги з боку фахівця.

Самостійна навчальна діяльність студента може здійснюватися через:

- запам'ятовування певної інформації за допомогою уважного прослуховування і конспектування лекцій, активної роботи під час практичних занять;
- роботу над конспектами лекцій, планами практичних занять;
- опрацювання літературних джерел (конспектування самостійно вивченого матеріалу, реферування);
- роботу з каталогами звичайних і електронних бібліотек, інформаційно-пошуковими сервісами *Internet*;
- вивчення навчального матеріалу за паперовими та електронними підручниками, навчальними посібниками, практикумами тощо;
- опрацювання матеріалу за першоджерелами, науковою і спеціальною літературою;
- підготовку доповідей, рефератів, написання курсових робіт, пошукову і науково-дослідну діяльність;
- самотестування.

Самостійна робота студента під час лекції. Лекційний матеріал призначається для найбільш раціонального спрямування студентів при вивченні навчальної дисципліни і акцентуванні уваги на її найбільш складних, вузлових питаннях. Належне ведення конспекту під час лекції сприяє збереженню необхідної інформації та дає студенту змогу в подальшому проаналізувати її. За умови подання лекційного матеріалу в усній формі засвоюється до 20 % інформації. Викладання інформатики в комп'ютерних класах або аудиторіях, обладнаних мультимедійним обладнанням (наприклад, мультимедійним проектором або сенсорним екраном), одночасно з демонстрацією студентам прийомів роботи з інтерфейсом програми користувача дає можливість підвищити рівень засвоєння лекційного матеріалу до 50–60 %.

Робота над конспектами лекцій, планами практичних занять. При підготовці до практичних занять студент має спиратися на складений ним конспект лекції. При опрацюванні матеріалу лекції слід зіставити законспектований матеріал з планом практичного заняття, що міститься у методичних матеріалах для практичних занять або у навчально-методичному комплексі. Якщо у конспекті бракує матеріалу з окремих питань лекції чи недостатньо розкриті деякі питання практичного заняття, або їх винесено на самостійне опрацювання, студент повинен звернутися до рекомендованих підручників, навчальних посібників і відповідних методичних матеріалів. Готуватися до практичного заняття краще за все з використанням ПЕОМ зі встановленим на ній відповідним програмним забезпеченням.

Вивчення навчального матеріалу за підручниками, навчальними посібниками, методичними вказівками, опрацювання матеріалу за першоджерелами, науковою і спеціальною літературою. Працювати з підручниками, навчальними посібниками, методичними вказівками, практикумами, науковою і спеціальною літературою незалежно від типу їхнього носія (паперового чи електронного) необхідно у такий спосіб, щоб здобути максимум теоретичних знань і набути практичних навичок. При роботі з цими джерелами студент насамперед повинен ознайомитися з їх змістом, щоб визначити, чи необхідно опрацювати певне джерело і чи має воно стосунок до навчального курсу, що вивчається, і тільки після цього визначити послідовність його опрацювання і відібрати з нього необхідний для вивчення матеріал (глави, розділи тощо). В разі роботи з інтерактивними електронними джерелами слід використовувати можливості навігації за документом, що надаються сучасними програмами, призначеними для читання електронних документів відповідних форматів (*MS Word, Adobe Reader, Adobe Acrobat* та ін.), і особливо переваги гіпертекстової технології подачі навчального матеріалу, а саме — за допомогою гіперпосилань знаходити відповіді на поставлені питання. При опрацюванні матеріалу необхідно з'ясувати суть питання, що вивчається, не уникаючи при цьому визначення змісту незрозумілих чи незнайомих слів і термінів. Саме інтерактивні гіпертекстові електронні джерела (довідки в складі програмних продуктів, електронні посібники та словники) дають змогу конкретизувати терміни та визначення якнайшвидше. При вивченні матеріалу необхідно аналізувати прочитане, порівнюючи з прослуханою та законспектованою лекцією, робити логічні висновки, позначати незрозумілі положення з метою їх подальшого з'ясування на практичному занятті. Бажано відпрацювати зручну для себе пев-

ну систему позначень (позначки на полях конспекту, підкреслення маркерами різних кольорів, доповнення конспекту альтернативними формулюваннями та посиланнями на інші джерела тощо) та фіксації опрацьованого матеріалу. Сучасні текстові редактори (передусім *MS Word*) надають можливість створювати електронний конспект з примітками, виносками, коментарями та його роздруківки. Для самостійного поглибленого вивчення навчального матеріалу студенту слід звертатися до наукової та спеціальної літератури, яка може бути і не зазначена в навчально-методичному комплексі. Використання самостійно отриманих відомостей як у навчанні, так і в практичній роботі є, безперечно, цінним здобутком діяльності студента на шляху формування професійного потенціалу.

Робота з бібліотечними фондами та дистанційними джерелами з метою пошуку необхідної інформації. Знання основ системного програмування та операційних систем належать до базової підготовки сучасного спеціаліста з інформаційних технологій. З позицій випереджаючої освіти навчання тільки за конспектом лекцій і основною літературою, наведеною у навчальній програмі, є недостатнім. У більшості випадків належна підготовка потребує вміння швидко знайти та опрацювати необхідний матеріал за першоджерелами, науковою і спеціальною літературою та коректно цитувати знайдене. Перелік такої літератури, як правило, наводиться у навчально-методичному комплексі навчальної дисципліни. Тому завдання студента зводиться до самостійного знаходження цих матеріалів шляхом пошуку у паперових або електронних фондах бібліотек, а також у різноманітних файлових архівах, базах даних і базах знань, доступ до яких здійснюється за допомогою відповідних сервісів *Internet* (в основному *Word Wide Web*, *FTP* та *UseNet newsgroups*).

Для пошуку документа використовуються різні його ознаки. Насамперед це реквізити документа (УДК. Автор(и). Заголовок опису. Основний заголовок: відомості, що відносяться до заголовку/Відомості про відповідальність. — Відомості про видання (в тому числі *URL* — адреса *Web*-документа або *Ftp*-файла). — Місце видання, дата видання. — Обсяг). УДК — це універсальна десяткова класифікація будь-яких офіційних видань по всьому світу. Відповідні довідники видаються багатьма мовами і постійно оновлюються. В Україні у 2006 р. Книжковою палатою України ім. І. Федорова видано “Універсальну десяткову класифікацію. Зміни та доповнення” (вип. 4) у паперовому варіанті. Довідкова база УДК постійно нарощується за рахунок електронних видань. Знання УДК дає можливість швидко

знайти необхідне джерело за систематичним бібліотечним каталогом. Наприклад, УДК видань з інформаційних технологій починається з 004.

Коли код УДК невідомий, то необхідно звернутися до алфавітного каталогу бібліотеки і за назвою джерела або прізвищем та ініціалами автора знайти відповідний бібліотечний шифр джерела.

Якщо ж студент здійснює наукове дослідження вибраної проблеми, готує наукову доповідь або виступ на конференції, і йому не відомі реквізити джерела або саме джерело, то слід зробити пошук у систематичному бібліотечному каталозі. Завдання студента полягає у пошуку необхідної галузі (підгалузі), що охоплює розшукувану інформацію, а потім у межах цієї галузі (підгалузі) — картки з необхідним джерелом і бібліотечним шифром. У подальшому студент повинен оформити бібліотечне замовлення на літературу встановленого зразка, до якого внести шифр знайденого джерела та усі необхідні реквізити. Робота з електронними фондами в цьому варіанті значно ефективніша, оскільки у розвинених бібліотеках облік літератури ведеться в середовищах систем управління базами даних, за допомогою яких пошук потрібної інформації здійснюється найефективніше.

Сервіси мережі *Internet* надають унікальні можливості знаходження літературних джерел у географічно віддалених фондах та архівах, а також шляхом участі у мережних конференціях, де можна отримати відповіді та поради щодо розшукуваної інформації. Для доступу до *Internet*-ресурсів необхідно знати їх мережну адресу. Оскільки *Internet* постійно оновлюється і розвивається, в ньому немає єдиного каталогу, змісту або наочного покажчика ресурсів. Проте в *Internet* існують різні інформаційно-пошукові системи, що допомагають користувачам знайти те, що їм потрібно. Це насамперед тематичні каталоги і так звані пошукові машини. Тематичні (наочні) каталоги — це інформаційно-довідкові системи, підготовлені вручну редакторами цих систем на основі інформації, зібраної на серверах *Internet*. Інформація в цих системах розподіляється за тематичними розділами відповідно до певної ієрархії. На верхньому рівні розділів зібрані загальні категорії (наприклад, “Інтернет”, “Бізнес”, “Мистецтво”, “Освіта” тощо), а нижній рівень складають посилання на конкретні *Web*-сторінки або інші інформаційні ресурси. Для швидкого переходу до потрібного розділу тематичного каталогу можна скористатися вбудованою системою автоматичного пошуку за ключовими словами. Для цього в рядку запити слід ввести ключове слово (поєднання слів),

клацнути **Пошук**, і система повідомить, чи є відповідний розділ в її каталозі, і запропонує в нього перейти, минаючи всі проміжні розділи. Рекомендуємо використовувати каталоги: <http://www.yahoo.com>, <http://www.portal.edu.ru>, <http://www.ipl.org>.

Пошукові системи є складними інформаційно-довідковими системами, що автоматично генеруються на основі даних, які збираються мережними програмами-роботами по всій *Internet*, і надають у відповідь на запит користувача посилання на різні *Internet*-ресурси. Запит здійснюється за певною процедурою (певною мовою), яка може відрізнятися в різних системах, проте в спрощеному вигляді вона полягає в тому, що користувач вводить в спеціальному полі (або в кількох полях) ключові слова та/або словосполучення, що найточніше відображають суть проблеми.

До загальних положень мов запитів належать такі:

- ключові слова можна вводити у відповідне поле пошукової системи поодиноці, послідовно звужуючи пошук, або ж вводити відразу кілька слів, розділяючи їх пробілами або комами. Регістр не має значення;
- режим пошуку “AND” (“І”) означає, що буде знайдено тільки ті дані, де зустрічається кожне з ключових слів;
- при використанні режиму “OR” (“АБО”) результатом пошуку будуть всі дані, де зустрічається хоча б одне з ключових слів;
- використовуйте знаки “+” і “-” перед ключовим словом. Щоб виключити документи, де зустрічається певне слово, поставте перед ним мінус. І навпаки, щоб певне слово обов’язково було присутнє в документі, поставте перед ним плюс. Зверніть увагу на те, що між знаком і словом не повинно бути пропуску;
- якщо ви хочете виключити якесь слово з пошуку, поставте перед ним знак “-”, наприклад, “+захист -Excell”;
- за замовчуванням програма шукає всі дані, де зустрічається введене вами слово. Наприклад, при запиті “редактор” будуть знайдені слова “редактор”, “текстовий”, “графічний”, “газети”, “головний” і багато інших. Знак оклику перед або після ключового слова означає, що будуть знайдені тільки слова, які точно відповідають запиту (наприклад, “текстовий! редактор!”).

Також корисно запам’ятати і використовувати при пошуку такі прийоми:

- якщо для пошуку потрібно ввести словосполучення, візьміть його в лапки;

- якщо ви пишете все слово малими буквами, буде знайдено всі варіанти його написання; якщо ви вказали хоча б одну букву в шуканому слові великою, то система шукатиме тільки такі варіанти.
- Якщо Ви хочете знайти не текст, а яке-небудь зображення, то можна користуватися словом *image*. Наприклад, *image: sea* дасть список сторінок із зображенням моря;
- якщо слово, яке Ви шукаєте, зустрічається в різних контекстах, можна виключити слова, які зустрічаються в непотрібному контексті. Наприклад, вказати аргумент пошуку *+Celeron +Price +UA -USA*;
- перевіряйте орфографію. Якщо пошук не дав результатів, можливо, при введенні ви зробили помилку.
- Використовуйте синоніми. Якщо список знайдених сторінок дуже малий або не містить корисних сторінок, спробуйте змінити слово. Наприклад, замість “реферати”, можливо, більше підійде “курсові роботи” або “твори”;
- якщо один із знайдених документів ближче до потрібної теми за інші, клацніть ***Знайти схожі документи***. Це посилання розташовано під короткими описами знайдених документів. Система проаналізує сторінку і знайде документи, схожі на ті, що ви зазначили.

Таких систем в *Internet* значно більше, ніж тематичних каталогів. Серед пошукових систем існують як обширні за тематикою метапошукові системи, так і вузькоспеціалізовані. Найвідоміші з них такі: <http://www.google.com>, <http://www.altavista.com>, <http://www.askjeeves.com>, <http://www.lycos.com>, <http://www.sciseek.com>, <http://www.msn.com>, <http://meta.ua>, <http://www.rambler.ru>, <http://www.yandex.ru>, <http://www.aport.ru>, <http://www.metabot.ru>, <http://newsgroups.langenberg.com>, uk.wikipedia.org, www.bukinist.agava.ru.

Матеріали щодо методів підвищення ефективності пошуку інформації в *Internet* містяться у статтях: <http://www.yandex.ru/info/search.html>, <http://www.searchengines.ru/>, <http://www.zodchiy.ru/links/search/>, <http://www.citforum.ru/internet/search/index.shtml>, <http://websearch.report.ru/>, <http://www.kokoc.com/search-engines/index.shtml>, <http://www.zhurnal.ru/search-r.shtml>

Самостійна робота має такі складові і форми їх оцінювання:

- підготовка та власне аудиторна робота під час практичних і лабораторних занять. Результати її оцінюються під час поточного контролю;

- виконання самостійних робіт у формі есе, рефератів з конкретних проблем та складання письмових звітів на електронних чи паперових носіях або у вигляді усних доповідей;
- опрацювання програмного матеріалу зі змістового модуля та оцінка його результатів під час проміжного контролю;
- виконання письмової контрольної роботи або тестування;
- звіт про проходження практики;
- звіт про науково-дослідну роботу, результати якої можуть бути використані при написанні випускної роботи і за рішенням кафедри опубліковані.

МЕТОДИЧНІ ВКАЗІВКИ ДО ПІДГОТОВКИ, НАПИСАННЯ ТА ЗАХИСТУ РЕФЕРАТУ

Реферат є складовою вивчення навчальної дисципліни.

Наведені завдання підготовлені відповідно до курсу “Сучасні комп’ютерні технології об’єктно-орієнтованого програмування на C++” для бакалаврів.

Мета — допомогти студентам засвоїти теоретичні знання в галузі сучасних комп’ютерних технологій об’єктно-орієнтованого програмування на C++. Оформлення й захист рефератів повинні сприяти активному засвоєнню нового матеріалу, виробленню у студентів уміння комплексно використовувати суміжні дисципліни при вирішенні практичних питань.

Структура реферату

Орієнтовна структура і обсяги реферату наведені у таблиці.

План (розділи)	Обсяг у сторінках (приблизно)	Короткий зміст (що потрібно висвітлити)
Вступ	До однієї	Мета, загальна характеристика, зазначення номера варіанта завдання
Назва кожного питання відповідно до теми реферату	1 – 2, загальний обсяг роботи у межах 20–30	Викладення суті питання з наведенням прикладів і посилань на літературні джерела
Висновки	До однієї	Прикладне значення
Список літератури	До однієї	
Додатки	До трьох	Якщо є

Загальний обсяг роботи не повинен перевищувати 20–30 сторінок машинописного тексту, надрукованого через 2 інтервали, рукописне викладення тексту не повинно перевищувати 18–24 сторінок шкільного зошита.

Виконання та оформлення реферату

Студент повинен виконати реферат, розкривши історичні та технологічні передумови цієї проблеми, відповісти на всі питання теоретичного плану, а також описати технологію розв'язання практичної задачі, якщо така передбачена рефератом.

Відповіді на теоретичні питання потребують ретельної роботи з літературою. Крім виписок і конспектування літературних джерел, наприклад, з *Internet*, студент повинен зробити висновки. Роботу треба виконати самостійно. В тексті реферату потрібно давати посилання на використану літературу. У висновках до реферату розглядають питання економічної доцільності і практичного застосування сучасних інформаційних технологій та обчислювальної техніки в сфері операційних систем та системного програмування.

Реферат слід оформляти на стандартних аркушах паперу, зброшурованих у папку. Усі сторінки мають бути пронумеровані. На титульній сторінці необхідно зазначити назву вищого навчального закладу, факультет, спеціальність, дисципліну, курс, групу, а також прізвище, ініціали та номер залікової книжки.

На першій сторінці потрібно навести представлені розрахунок варіанта контрольної роботи та питання варіанта і проставити номери сторінок, на яких викладено цей матеріал. На останній сторінці студент підписує роботу і ставить дату. Наприкінці роботи необхідно подати список використаної літератури. Зшита папка повинна бути вкладена в поліетиленовий файл та містити дискету з повним текстом, графікою і т. ін. набраного варіанта реферату.

Вибір варіанта реферату

Кожний студент отримує окреме завдання для виконання КР згідно з варіантом Z , котрий обчислюється за формулою:

$$Z = \text{mod}_{10}(NZK + PR - 2000) + 1,$$

де NZK — номер залікової книжки (студентського квитка) студента; PR — поточний рік отримання завдання.

Наприклад, $NZK = 398$, $PR = 2002$, тоді

$$Z = \text{mod}_{10} (398 + 2002 - 2000) + 1 = \text{mod}_{10} (400) + 1 = 1.$$

Отже, $Z = 1$.

Зауваження. Обчислення варіанта поотрібно навести у вступі до контрольної роботи.

Для довідки: $\text{mod}_a b$ дорівнює залишку від ділення b на a .

Увага!

Неправильно оформлена робота повертається без перевірки на дооформлення. Робота, виконана не за своїм варіантом, підлягає переробці

ІНДИВІДУАЛЬНО-КОНСУЛЬТАЦІЙНА РОБОТА

Індивідуально-консультаційна робота з дисципліни здійснюється у формі консультацій за графіком (одна консультація на два тижні). На консультаціях студентам надаються пояснення щодо виконання самостійної роботи, підготовки до практичних занять, перевірки та захисту завдань, винесених на поточний контроль тощо.

Мета вивчення дисципліни:

- оволодіти комплексом знань із сучасних комп'ютерних технологій об'єктно-орієнтовного програмування. Вивітлити основні теоретичні концепції об'єктно-орієнтованого програмування, а також реалізацію у мові програмування C++;
- опанувати сучасні методи та алгоритми проектування об'єктно-орієнтованих програм і їх реалізацію мовою C++;
- набути навичок проектування власних ієрархічних об'єктно-орієнтованих бібліотек та навчитися використовувати вже існуючі для побудови різноманітних програм. Здобути знання з основних об'єктно-орієнтованих бібліотек мови C++.

У результаті самостійного вивчення навчальної дисципліни “Сучасні комп'ютерні технології об'єктно-орієнтованого програмування на C++” студент повинен:

- знати головні концептуальні особливості об'єктно-орієнтованого програмування, а також його реалізацію мовою C++;
- вміти проектувати та програмувати прикладні застосування з використанням об'єктно-орієнтованої технології програмування

ня. Набути навичок аналізу об'єктно-орієнтованого коду та методів його оптимізації;

- вміти проектувати власні ієрархічні об'єктно-орієнтовані бібліотеки.

ТЕМИ І ПЛАНИ САМОСТІЙНОЇ РОБОТИ СТРУКТУРА ТА ФОРМА ЗВІТНОСТІ

№ теми	Назва змістового модуля і теми	Зміст завдання	Форми контролю
1	2	3	4
Змістовий модуль І. Основи об'єктно-орієнтованого програмування. Реалізація об'єктно-орієнтованого програмування у мові С++			
1	Основні поняття об'єктно-орієнтованого програмування	1. Інкапсуляція, наслідування, поліморфізм. 2. Переваги об'єктно-орієнтованого програмування	Конспект
2	Класи, реалізація інкапсуляції в мові в С++	1. Опис класів та об'єктів у С++. 2. Модифікатори доступу до членів класів. 3. Методи класів. Скрите поле <i>this</i> . 4. Конструктори. Конструктори за замовчуванням. 5. Статичні елементи класів. 6. Константні поля і методи. 7. Дружні функції та класи. 8. Деструктори. 9. Перевантаження операцій. 10. Рекомендації щодо проектування класів	Конспект
3	Наслідування, батьківські та дочірні класи	1. Види наслідування. 2. Просте наслідування. 3. Ідентифікатори доступу при наслідуванні 4. Множинне наслідування.	

1	2	3	4
4	Поліморфізм, перевантаження методів	<ol style="list-style-type: none"> 1.Перевантаження методів. 2.Раннє та пізнє зв'язування. 3.Віртуальні методи, перевантаження віртуальних методів. 4.Абстрактні класи, чисто віртуальні методи 	Конспект
5	Шаблони класів, абстрактне програмування	<ol style="list-style-type: none"> 1.Шаблони методів. 2.Створення та використання шаблонів класів. 3.Спеціалізація шаблонів класу. 4.Абстрактне програмування, переваги та недоліки абстрактного програмування 	Конспект
6	Виключення, обробка виключень у мові C++	<ol style="list-style-type: none"> 1.Механізм обробки та синтаксис виключень. 2.Перехват виключень. 3.Ієрархія виключень. 4.Загальні рекомендації щодо використання виключень 	Конспект
7	Перетворення типів	<ol style="list-style-type: none"> 1.Загальні відомості про перетворення типів. 2.Явне та неявне перетворення типів. 3.Операція <i>const_cast</i>. 4.Операція <i>dynamic_cast</i>. 5.Операція <i>static_cast</i> 	Конспект
Змістовий модуль II. Стандартні бібліотеки системи програмування мови C++			
1	Потокові класи	<ol style="list-style-type: none"> 1.Стандартні потоки. 2.Форматування даних, флаги і форматуючі методи. 3.Маніпулятори, прості та параметризовані маніпулятори. 4.Методи обміну з потоками. 5.Помилки потоків. 6.Файлові потоки. 7.Строкові потоки. 8.Потоки і типи, визначені користувачами 	Конспект

1	2	3	4
2	Рядки	<ol style="list-style-type: none"> 1. Конструктори і присвоювання рядків. 2. Операції над рядками: об'єднання, перетворення, пошук, порівняння. 3. Отримання характеристик рядків 	
3	Контейнерні класи	<ol style="list-style-type: none"> 1. Загальні відомості про контейнерні класи і контейнери. 2. Послідовні контейнери. 3. Вектори. 4. Двосторонні черги. 5. Списки. 6. Стеки. 7. Черги з пріоритетами. 8. Асоціативні контейнери. 9. Словники. 10. Словники з дублікатами. 11. Множини. 12. Множини з дублікатами. 13. Бітові множини 	
4	Ітератори і функціональні об'єкти	<ol style="list-style-type: none"> 1. Літератори. 2. Обернені оператори. 3. Літератори вставки. 4. Потоківі літератори. 5. Функціональні об'єкти. 6. Арифметичні функціональні об'єкти. 7. Предикати, віднімателі, зв'язувачі. 8. Адаптери вказівників на функції. 9. Адаптери методів 	
5	Алгоритми	<ol style="list-style-type: none"> 1. Немодифікуючі операції з послідовностями. 2. Модифікуючі операції з послідовностями. 3. Алгоритми, пов'язані з сортуванням. 4. Алгоритми роботи з множинами і пірамідами 	
6	Засоби для здійснення числових розрахунків	<ol style="list-style-type: none"> 1. Узагальнені числові алгоритми. 2. Клас <i>valarray</i>. 3. Допоміжні класи для здійснення числових розрахунків 	

Реферат за змістовим модулем II

Теми рефератів за змістовим модулем І

1. Сучасний стандарт мови програмування C++ ISO/IEC 14882.
Література [1; 2; 7]
2. Реалізація технології об'єктно-орієнтованого програмування у мові програмування C++.
Література [6; 9–11]
3. Основні ідеї технології об'єктно-орієнтованого програмування.
Література [1; 2; 5; 8; 10; 11]
4. Конструктори: види, опис і способи використання.
Література [1; 2; 5; 8; 10; 11]
5. Деструктори: види, опис і способи використання.
Література [1; 2; 5; 8; 10; 11]
6. Модифікація об'єктів з допомогою перевантаження операцій.
Література [5; 8; 10; 11; 12; 18]
7. Використання перетворення типів у сучасному програмуванні на C++.
Література [1; 5; 8; 10;]
8. Використання виключень для опрацювання помилкових ситуацій у мові програмування C++.
Література [1–3; 5; 8–10]
9. Сучасні методології програмування на мові C++.
Література [1–5; 8; 10; 16]
10. Методи та засоби проектування класів та ієрархічних об'єктно-орієнтованих бібліотек на мові C++.
Література [2; 5; 8–11]

Питання для самоконтролю і співбесіди за змістовим модулем І

1. Які відмінності між процедурним, об'єктним та об'єктно-орієнтованим програмуванням?
2. Що таке директиви препроцесора? Яке їх призначення?
3. Як можна виділити та звільнити область пам'яті для вказівника?
4. Охарактеризуйте особливості передачі параметрів функції. Опишіть можливості використання значення параметрів за замовчуванням.
5. Визначення і оголошення глобальних об'єктів і функцій.
6. Що таке конструктори та деструктори. Яке їх призначення і сфера застосування?

7. Область видимості і права доступу до функцій класу.
8. Ідентичність об'єктів, копіювання, ініціалізація і присвоєння об'єктів.
9. Властивості конструкторів, реалізація кількох конструкторів у класі, конструктор за замовчуванням.
10. Ієрархія об'єктів і ієрархія класів. Головні принципи побудови ієрархічних бібліотек.
11. Захищена область класу.
12. Послідовність виклику конструкторів.
13. Як реалізується раннє і пізнє зв'язування?
14. Назвіть ознаки складних систем.
15. Охарактеризуйте елемент об'єктної моделі — інкапсуляцію.
16. У чому переваги об'єктної моделі над процедурною?

Завдання за змістовим модулем I

Завдання 1

Описати клас, що реалізовує стек. Написати програму, що використовує цей клас для моделювання T-подібного сортувального вузла на залізниці. Програма повинна розділяти на два напрями потяг, що складається з вагонів двох типів (на кожен напрям формується потяг з вагонів одного типу). Передбачити можливість формування потяга з файла і з клавіатури.

Завдання 2

Описати клас, що реалізовує бінарне дерево, яке надає можливість додавання нових елементів, видалення тих, що існують, пошуку елемента за ключем, а також послідовного доступу до всіх елементів. Написати програму, що використовує цей клас для представлення англо-російського словника. Програма повинна містити меню, яке дає змогу перевірити всі методи класу. Передбачити можливість формування словника з файла і з клавіатури.

Завдання 3

Побудувати систему класів для опису плоских геометричних фігур: кола, квадрата, прямокутника. Передбачити методи для створення об'єктів, переміщення на площині, зміни розмірів і обертання на заданий кут. Написати програму, що демонструє роботу з цими кла-

сами. Програма повинна містити меню, що дає можливість перевірити всі методи класів.

Завдання 4

Побудувати опис класу, що містить інформацію про поштову адресу організації. Передбачити можливість роздільної зміни складових частин адреси, створення і знищення об'єктів цього класу. Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дає змогу перевірити всі методи класу.

Завдання 5

Побудувати клас для представлення комплексних чисел. Забезпечити виконання операцій складання, віднімання і множення комплексних чисел. Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дає можливість перевірити всі методи класу.

Завдання 6

Побудувати клас для об'єктів-векторів, що задаються координатами кінців у тривимірному просторі. Забезпечити операції складання і віднімання векторів з отриманням нового вектора (суми або різниці), обчислення скалярного добутку двох векторів, довжини вектора, косинуса кута між векторами. Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дає змогу перевірити всі методи класу.

Завдання 7

Побудувати клас прямокутників із сторонами, паралельними осям координат. Передбачити можливість переміщення прямокутників на площині, зміна розмірів, побудова найменшого прямокутника, що містить два задані прямокутники, і прямокутники, що є загальною частиною (перетином) двох прямокутників. Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дає можливість перевірити всі методи класу.

Завдання 8

Побудувати клас для визначення одновимірних масивів цілих чисел (векторів). Передбачити можливість звернення до окремого елемента масиву з контролем виходу за межі масиву, можливість вста-

новлення довільних меж індексів при створенні об'єкта, можливість виконання операцій поелементного додавання і віднімання масивів з однаковими межами індексів, множення і ділення всіх елементів масиву на скаляр, виведення на екран елемента масиву за заданим індексом, виведення на екран усього масиву. Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дає змогу перевірити всі методи класу.

Завдання 9

Побудувати клас для визначення одновимірних масивів рядків фіксованої довжини. Передбачити можливість звернення до окремих рядків масиву за індексами, контроль виходу за межі масиву. Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дає можливість перевірити всі методи класу.

Завдання 10

Побудувати клас многочленів від однієї змінної, що задаються степенем многочлена і масивом коефіцієнтів. Передбачити методи для обчислення значення многочлена для заданого аргументу, операції складання, віднімання і множення многочленів з отриманням нового об'єкта-многочлена, виведення на екран опису многочлена. Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дає змогу перевірити всі методи класу.

Завдання 11

Побудувати клас одновимірних масивів рядків, кожен рядок задається завдовжки і покажчиком на виділену для неї пам'ять. Передбачити можливість звернення до окремих рядків масиву за індексами, контроль виходу за межі масивів, виконання операцій поелементного додавання двох масивів з утворенням нового масиву, злиття двох масивів з виключенням елементів, що повторюються, вивід на екран елемента масиву і всього масиву. Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дає змогу перевірити всі методи класу.

Завдання 12

Побудувати клас, що забезпечує представлення матриці довільного розміру з можливістю зміни числа рядків і стовпців, виведення на екран підматриці будь-якого розміру і всієї матриці. Написати про-

граму, що демонструє роботу з цим класом. Програма повинна містити меню, що дає змогу перевірити всі методи класу.

Завдання 13

Написати клас для ефективної роботи з рядками, що дає можливість формувати і порівнювати рядки, зберігати в рядках числові значення і витягувати їх. Для цього необхідно реалізувати:

- перевантажені операції присвоювання і конкатенації;
- операції порівняння і зведення типів;
- перетворення в число будь-якого типу;
- виведення формату рядка.

Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дає змогу перевірити всі методи класу.

Завдання 14

Описати клас “домашня бібліотека”. Передбачити можливість роботи з довільною кількістю книжок, пошуку книги за якоюсь ознакою (наприклад, за автором або за роком видання), додавання книжок у бібліотеку, видалення книжок з неї, сортування книжок за різними полями. Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дає можливість перевірити всі методи класу.

Завдання 15

Описати клас “записник”. Передбачити можливість роботи з довільним числом записів, пошуку запису за якоюсь ознакою (наприклад, за прізвищем, датою народження або номером телефону), додавання і видалення записів, сортування за різними полями. Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дає змогу перевірити всі методи класу.

Завдання 16

Описати клас “студентська група”. Передбачити можливість роботи змінною кількістю студентів, пошуку студента за якоюсь ознакою (наприклад, за прізвищем, датою народження або номером телефону), додавання і видалення записів, сортування за різними полями. Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дає можливість перевірити всі методи класу.

Завдання 17

Описати клас, що реалізовує тип даних “матриці” і роботу з ними. Клас повинен реалізовувати такі операції над матрицями:

- додавання, віднімання, множення, ділення ($+$, $-$, $*$, $/$) (множення і ділення як на іншу матрицю, так і на число);
- комбіновані операції присвоювання ($+=$, $-=$, $*=$, $/=$);
- операції порівняння типу рівність/нерівність;
- операції обчислення оберненої і транспонованої матриці, операцію піднесення до ступеня;
- методи обчислення детермінанта і норми;
- методи, що реалізують перевірку типу матриці (квадратна, діагональна, нульова, одинична, симетрична, верхня трикутна, нижня трикутна);
- операції введення/виведення в стандартні потоки.

Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дає змогу перевірити всі методи класу.

Завдання 18

Описати клас “множина”, що дає можливість виконувати основні операції — додавання і видалення елемента, перетин, об’єднання і різниця множин. Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дає змогу перевірити всі методи класу.

Завдання 19

Описати клас, що реалізовує стек. Написати програму, що використовує цей клас для відшукування виходу з лабіринту. Лабіринт подається у вигляді матриці, що складається з квадратів. Кожен квадрат або відкритий, або закритий. Вхід у закритий квадрат заборонений. Якщо квадрат відкритий, то вхід у нього можливий з боку, але не з кута. Кожен квадрат визначається його координатами в матриці. Після відшукування проходу програма друкує знайдений шлях у вигляді координат квадратів.

Завдання 20

Описати клас “наочний покажчик”. Кожен компонент покажчика містить слово і номери сторінок, на яких це слово зустрічається.

Кількість номерів сторінок, що відносяться до одного слова, від одного до десяти. Передбачити можливість формування покажчика з клавіатури і з файла, виведення покажчика виведення номерів сторінок для заданого слова, видалення елемента з покажчика. Написати програму, що демонструє роботу з цим класом. Програма повинна містити меню, що дає змогу перевірити всі методи класу.

Загальна частина для завдань 21–25

Написати програму, що демонструє роботу з об'єктами двох типів — T1 і T2, для чого створити систему відповідних класів. Кожен об'єкт повинен мати ідентифікатор (у вигляді довільного рядка символів) і одне або кілька полів для зберігання стану об'єкта (один клас є нащадком іншого). Клієнтові (функції main) повинні бути доступні такі основні операції (методи): створити об'єкт, видалити об'єкт, показати значення об'єкта та інші додаткові операції (залежать від варіанта). Операції зі створення і видалення об'єктів інкапсулювати в класі Factory. Передбачити меню, що дає можливість продемонструвати задані операції. За необхідності в класи, що розробляються, додаються додаткові методи (наприклад, конструктор копіювання, операція привласнення і т. ін.) для забезпечення належного функціонування цих класів.

Завдання 21

1. Визначити клас з ім'ям STUDENT, що містить такі поля: прізвище і ініціали; номер групи; успішність (масив з п'яти елементів).

Визначити методи доступу до цих полів і перевантажені операції видалення і вставки для об'єктів типу STUDENT.

2. Написати програму, що виконує такі дії: виведення з клавіатури даних у масив, що складається з десяти об'єктів типу STUDENT; записи повинні бути впорядковані за збільшенням номера групи; виведення на дисплей прізвищ і номерів груп для всіх студентів, включених у масив, якщо середній бал студента перевищує 4.0; якщо таких студентів немає, вивести відповідне повідомлення.

Завдання 22

1. Визначити клас з ім'ям AEROFLOT, що містить такі поля: назва пункту призначення рейса, номер рейсу; тип літака.

Визначити методи доступу до цих полів і переобтяжені операції витягання і вставки для об'єктів типу AEROFLOT.

2. Написати програму, що виконує такі дії: введення з клавіатури даних у масив, що складається з семи об'єктів типу AEROFLOT; записи повинні бути впорядковані за збільшенням номера рейсу; виведення на екран номерів рейсів і типів літаків, що вилітають у пункт призначення, назва якого збіглася з назвою, введеною з клавіатури; якщо таких рейсів немає, видати на дисплей відповідне повідомлення.

Завдання 23

1. Визначити клас з ім'ям WORKER, що містить такі поля: прізвище і ініціали працівника; назва посади; рік приходу на роботу. Визначити методи доступу до цих полів і переобтяжені операції витягання і вставки для об'єктів типу WORKER.

2. Написати програму, що виконує такі дії: введення з клавіатури даних у масив, що складається з десяти об'єктів типу WORKER; записи повинні бути розміщені за абеткою; виведення на дисплей прізвищ працівників, чий стаж роботи в організації перевищує значення, введене з клавіатури; якщо таких працівників немає, вивести на дисплей відповідне повідомлення.

Завдання 24

1. Визначити клас з ім'ям TRAIN, що містить наступні поля: назва пункту призначення; номер потяга; час відправлення. Визначити методи доступу до цих полів і переобтяжені операції витягання і вставки для об'єктів типу TRAIN.

2. Написати програму, що виконує такі дії: введення з клавіатури даних в масив, що складається з восьми об'єктів типу TRAIN; записи повинні бути розміщені в алфавітному порядку за назвами пунктів призначення; виведення на екран інформації про потяги, що відправляються після введеного з клавіатури часу; якщо таких потягів немає, видати на дисплей відповідне повідомлення.

Завдання 25

1. Визначити клас з іменем MARSH, що містить такі поля: назву початкового пункту маршруту; назву кінцевого пункту маршруту;

номер маршруту. Визначити методи доступу до цих полів і переоб'яжені операції витягання і вставки для об'єктів типу MARSH.

2. Написати програму, що виконує такі дії: введення з клавіатури даних у масив, що складається з восьми об'єктів типу MARSH; записи повинні бути впорядковані за номерами маршрутів; виведення на екран інформації про маршрут, номер якого введений з клавіатури; якщо таких маршрутів немає, видати на дисплей відповідне повідомлення.

Питання для самоконтролю та співбесіди за змістовим модулем II

1. Що таке потік? Які основні алгоритми роботи з потоками реалізовано у C++?
2. Які переваги надають потоки у C++ порівняно з функціями введення-виведення в мові C?
3. Охарактеризуйте основні потокові класи `cin`, `cout`, `cerr`, `clog`. Наведіть приклади їх застосування.
4. Наведіть основні випадки виникнення помилок у потоках. Які засоби реалізовано у мові C++ для обробки помилок потоків введення-виведення?
5. Охарактеризуйте строкові потоки в C++.
6. Дайте коротку характеристику контейнерним класам.
7. Надайте класифікацію контейнерних класів та їх коротку характеристику.
8. Дайте означення лінійним спискам. Яке їх призначення, переваги та недоліки порівняно з іншими структурами даних?
9. Назвіть випадки, коли дані необхідно організувати в черги з пріоритетами?
10. Дайте коротку характеристику словникам, словникам з дублікатами, множинам, множинам з дублікатами і бітовим множинам.
11. Охарактеризуйте функціональні об'єкти і їх використання в бібліотеці TSL.
12. Перелічіть основні функції алгоритмів бібліотеки TSL, пов'язаних з сортуванням.
13. Які засоби обробки помилок при виділенні пам'яті реалізовані в стандартній бібліотеці C++?
14. Охарактеризуйте засоби роботи з комплексними числами стандартної бібліотеки C++.

Завдання за змістовим модулем II

Завдання 1

Написати програму, яка зчитує з текстового файлу три речення і виводить їх у зворотному порядку.

Завдання 2

Написати програму, яка зчитує текст з файлу і виводить на екран тільки речення, що містять задане з клавіатури слово.

Завдання 3

Написати програму, яка зчитує текст з файлу і виводить на екран тільки рядки, що містять двозначні числа.

Завдання 4

Написати програму, яка зчитує англійський текст з файлу і виводить на екран слова, що починаються з голосних букв.

Завдання 5

Написати програму, яка зчитує текст з файлу і виводить його на екран міняючи місцями кожні два сусідні слова.

Завдання 6

Написати програму, яка зчитує текст з файлу і виводить на екран тільки речення, що не містять ком.

Завдання 7

Написати програму, яка зчитує текст з файлу і визначає, скільки в ньому слів, що складаються не більш як з чотирьох букв.

Завдання 8

Написати програму, яка зчитує текст з файлу і виводить на екран тільки цитати, тобто речення, взяті в лапки.

Завдання 9

Написати програму, яка зчитує текст з файлу і виводить на екран тільки речення, що складаються із заданої кількості слів.

Завдання 10

Написати програму, яка зчитує англійський текст з файла і виводить на екран слова тексту, що починаються і закінчуються голосними буквами.

Завдання 11

Написати програму, яка зчитує текст з файла і виводить на екран тільки рядки, що не містять двозначних чисел.

Завдання 12

Написати програму, яка зчитує текст з файла і виводить на екран тільки речення, що починаються з тире, перед яким можуть бути тільки порожні символи.

Завдання 13

Написати програму, яка зчитує англійський текст з файла і виводить його на екран, замінивши кожну першу букву слів, що починаються з голосної букви, на велику.

Завдання 14

Написати програму, яка зчитує текст з файла і виводить його на екран, замінивши цифри від 0 до 9 на слова “нуль”, “один”, ..., “дев'ять”, починаючи кожне речення з нового рядка.

Завдання 15

Написати програму, яка зчитує текст з файла, знаходить найдовше слово і визначає, скільки разів воно зустрілося в тексті.

Завдання 16

Написати програму, яка зчитує текст з файла і виводить на екран спочатку питальні, а потім окличні речення.

Завдання 17

Написати програму, яка зчитує текст з файла і виводить його на екран, після кожного речення додаючи, скільки разів зустрілося в ньому задане з клавіатури слово.

Завдання 18

Написати програму, яка зчитує текст з файла і виводить на екран його речення в зворотному порядку.

Завдання 19

Написати програму, яка зчитує текст з файла і виводить на екран спочатку речення, що починаються з однобуквених слів, а потім решту.

Завдання 20

Написати програму, яка зчитує текст з файла і виводить на екран речення, що містять максимальну кількість знаків пунктуації.

Завдання 21

Написати програму для моделювання T-подібного сортувального вузла на залізниці з використанням контейнерного класу `stack` з STL. Програма повинна розділяти на два напрями склад, що складається з вагонів двох типів (на кожен напрям формується склад з вагонів одного типу). Передбачити можливість введення початкових даних з клавіатури і з файла.

Завдання 22

Написати програму, що знаходить прохід по лабіринту, з використанням контейнерного класу `stack` з STL. Лабіринт представляється у вигляді матриці, що складається з квадратів. Кожен квадрат або відкритий, або закритий. Вхід в закритий квадрат заборонений. Якщо квадрат відкритий, то вхід в нього можливий з боку, але не з гуги.

Програма знаходить прохід через лабіринт, рухаючись від заданого входу. Після відшукування виходу програма виводить знайдений шлях у вигляді координат квадратів.

Завдання 23

Написати програму, що моделює управління каталогом у файловій системі. Для кожного файла в каталозі містяться наступні відомості: ім'я файла, дата створення, кількість звернень до файла. Програма повинна забезпечувати:

- початкове формування каталогу файлів;
- виведення каталогу файлів;
- видалення файлів, дата створення яких раніше заданої;
- вибірку файла з найбільшою кількістю звернень.

Модельована функція повинна вибиратися за допомогою меню. Для подання каталогу використовувати контейнерний клас `list` з STL.

Завдання 24

Написати програму моделювання роботи автобусного парку. Відомості про кожен автобус містять: номер автобуса, прізвище і ініціали водія, номер маршруту. Програма повинна забезпечувати вибір за допомогою меню і виконання однієї з таких функцій:

- початкове формування даних про всі автобуси в парку у вигляді списку (введення з клавіатури або з файла);
- імітація виїзду автобуса з парку: вводиться номер автобуса; програма видаляє дані про цей автобус із списку автобусів, що є в парку, і записує ці дані в список автобусів, що знаходяться на маршруті;
- імітація в'їзду автобуса в парк: вводиться номер автобуса; програма видаляє дані про цей автобус із списку автобусів, що перебувають на маршруті і записує ці дані в список автобусів, що знаходяться в парку;
- виведення відомостей про автобуси, що знаходяться в парку, і про автобуси, що перебувають на маршруті.

Для подання необхідних списків використовувати контейнерний клас `list`.

Завдання 25

Написати програму обліку заявок на авіаквитки. Кожна заявка містить: пункт призначення, номер рейсу, прізвище і ініціали пасажера, бажану дату вильоту. Програма повинна забезпечувати вибір за допомогою меню і виконання однієї з таких функцій:

- додавання заявок у список;
- видалення заявок;
- виведення заявок за заданим номером рейсу і датою вильоту;
- виведення всіх заявок.

Для зберігання даних використовувати контейнерний клас `list`.

Завдання 26

Написати програму обліку книжок у бібліотеці. Відомості про книжки містять: прізвище і ініціали автора, назву, рік видання, кількість екземплярів даної книжки в бібліотеці. Програма повинна

забезпечувати вибір за допомогою меню і виконання однієї з таких функцій:

- додавання даних про книжки, що знову надходять у бібліотеку;
- видалення даних про списувані книжки;
- видача відомостей про всі книжки, впорядковані за прізвищами авторів;
- видача відомостей про всі книжки, впорядковані за роками видання.

Зберігання даних організувати із застосуванням контейнерного класу `multimap`. Як ключ використовувати “прізвище і ініціали автора”.

Завдання 27

Написати програму “Мій записник”. Передбачити можливість роботи з довільною кількістю записів, пошуку запису за якою-небудь ознакою (наприклад, за прізвищем, датою народження або номером телефону), додавання і видалення записів, сортування за різними полями. Зберігання даних організувати із застосуванням контейнерного класу `tar` або `multimap`.

Завдання 28

Написати програму “Автоматизована інформаційна система на залізничному вокзалі”. Інформаційна система містить дані про відправлення потягів далекого сполучення. Для кожного потяга зазначається: номер потяга, станція призначення, час відправлення. Програма повинна забезпечувати вибір за допомогою меню і виконання однієї з таких функцій:

- початкове введення даних в інформаційну систему (з клавіатури або з файла);
- виведення відомостей щодо всіх потягів;
- виведення відомостей щодо потяга із зазначеним номером;
- виведення відомостей щодо тих потягів, які слідують до зазначеної станції призначення.

Зберігання даних організувати із застосуванням контейнерного класу `vector`.

Завдання 29

Написати програму “Англо-російський і російський-англійський словник”. “База даних” словника повинна містити синонімічні

завдання перекладу слів. Програма повинна забезпечувати вибір за допомогою меню і виконання однієї з таких функцій:

- завантаження “бази даних” словника (з файла);
- вибір режиму роботи:
 - англо-російський;
 - російсько-англійський;
- виведення завдання перекладу заданого англійського слова.
- виведення завдання перекладу заданого російського слова.

Базу даних словника реалізувати у вигляді двох контейнерів типу тар.

Завдання 30

Написати програму, що розв’язує гру-головоломку “Гра в 15”. Початкове розміщення номерів — випадкове. Передбачити два режими демонстрації рішення: безперервний (з деякою затримкою візуалізації) і покроковий (при натисканні будь-якої клавіші). У програмі використовувати контейнерні класи STL.

Завдання 31

Скласти програму формування списку кандидатів, що беруть участь у виборах мера. Кожна заявка від кандидата містить: прізвище і ініціали, дату народження, місце народження, індекс популярності. Програма повинна забезпечувати вибір за допомогою меню і виконання однієї з таких функцій:

- додавання заявки в список кандидатів. Для введення індексу популярності (значення, вказане в дужках) передбачити вибір з допомогою підміну одного із зазначених завдань:
 - підтриманий президентом (70);
 - підтриманий опозиційною партією (15);
 - опозиційний кандидат, який зніме свою кандидатуру на користь кандидата № 1 (10);
 - інших (5);
- видалення заявки за заявою кандидата.

Зберігання даних організувати із застосуванням контейнерного класу `priority_queue` з STL. Для належного функціонування черги з пріоритетами потурбуватися про належне визначення операції `<` (менше) у класі, що описує заявку кандидата. Формування і виведення списку для голосування реалізувати за допомогою вибірки заявок з черги.

ПИТАННЯ ДЛЯ САМОКОНТРОЛЮ

1. Причини та передумови виникнення процедурного та модульного програмування.
2. Причини та передумови виникнення об'єктно-орієнтованого програмування.
3. Чому технологія об'єктно-орієнтованого програмування є однією з найпоширеніших?
4. Що таке препроцесор?
5. Що таке компілятор? Які особливості компілятора середовища візуального програмування Visual C++ 2005?
6. Які можливості надають директиви препроцесора?
7. Поясніть, як за допомогою директив препроцесора можна створювати апаратно-незалежні програми.
8. Що називають програмним модулем?
9. Опишіть способи ініціалізації змінних різних типів.
10. Опишіть найчастіше застосовувані способи використання вказівників.
11. Що таке структури? Для чого використовуються структури?
12. Опишіть призначення функцій та особливості їх оголошення.
13. Чому для передачі параметрів об'єктів великого розміру бажано використовувати вказівники?
14. Особливості повернення та використання значення функцій.
15. Що таке область видимості, область дії? Наведіть випадки, коли область видимості не збігається з областю дії.
16. Основні переваги та недоліки у використанні глобальних об'єктів.
17. Тривалість життя об'єктів: автоматичних, статичних і динамічних.
18. Охарактеризуйте основні принципи інкапсуляції даних та її реалізацію на мові C++.
19. Чому для ініціалізації полів класу найчастіше використовують конструктори?
20. Унаслідок чого може виникнути “засмічення” пам'яті? Як можна використати деструктори для розв'язання цих проблем?
21. Що таке класи, екземпляри класів, об'єкти? Які атрибути об'єктів “члени класів” ви знаєте?
22. Функції класу, функції поза класом.
23. Як можна отримати доступ до змінної, яка є в області видимості, але відсутня в області дії?

24. Що таке інтерфейс класу і реалізація класу? Які є способи реалізації класів?
25. Визначення і оголошення класу; вбудовані функції класу.
26. Як у C++ можна створювати двійників (клонів) об'єктів?
27. Скрите поле методів класу this. Як можна використовувати у методах класу цей вказівник?
28. Особливості створення і ініціалізації об'єктів на мові C++.
29. Порядок виконання конструкторів у дочірньому класі.
30. Ліквідація об'єктів, властивості деструкторів.
31. Наслідування, види наслідування, ідентифікатори доступу в наслідуванні.
32. Основні причини, якими керуються при проектуванні ієрархічних бібліотек.
33. Що називають базовим класом?
34. Що називають дочірнім класом?
35. Як у програмі можна використати захищену область класу?
36. Що таке область видимості класу?
37. Особливості створення та ліквідації дочірніх об'єктів.
38. Послідовність виклику конструкторів.
39. Деструктор дочірнього класу, послідовність виклику деструкторів.
40. Охарактеризуйте статичне і динамічне зв'язування.
41. Дайте коротку характеристику раннього і пізнього зв'язування.
42. Що таке віртуальні функції та віртуальні деструктори?
43. Що таке поліморфізм? Які переваги надає поліморфізм для побудови програм?
44. Чому більшість програм є складними системами?
45. Опишіть фази розробки програмного забезпечення.
46. Охарактеризуйте елемент об'єктної моделі — абстрагування.
47. Які переваги надає інкапсулювання даних?
48. Охарактеризуйте елемент об'єктної моделі — модульність.
49. Охарактеризуйте елемент об'єктної моделі — ієрархічність.
50. Чому об'єктна модель програм отримала ширше застосування, ніж процедурно-модульна?
51. Передумови виникнення концепції потоків у C++.
52. Призначення буфера в організації потокової взаємодії.
53. Стандартні базові класи, що стоять у голові потокової ієрархії.
54. Чому використання потоків у C++ для введення-виведення інформації є більш прийнятними, ніж функції введення-виведення в мові C?

55. Яке призначення класів `ios` та `streambuf`?
56. Яке призначення класів `ostream`, `istream` та `iostream`? Наведіть їх основні властивості та сфери застосування.
57. Які засоби надає мова `C++` для форматowanego введення даних? Поясніть, що таке прапорці та форматуючі методи.
58. Дайте означення маніпуляторів. Які види маніпуляторів є у мові `C++`?
59. Основні джерела помилок у потокових класах `C++`. Яких правил необхідно дотримуватися для уникнення помилок введення-виведення?
60. Охарактеризуйте файлові потоки у `C++`.
61. Призначення класів `fstream`, `ifstream` та `ofstream`. Які конструктори визначені в цих класах?
62. Які можливості надають строкові потоки порівняно з масивами символів `C++`?
63. Перелічіть основні алгоритми роботи зі строками в `C++` та стандартні засоби їх реалізації.
64. Охарактеризуйте строковий клас `string`.
65. Чому контейнерні класи отримали таку популярність у `C++`?
66. Призначення бібліотеки `STL`.
67. Які переваги надають контейнерні класи?
68. Коротко охарактеризуйте основні контейнерні класи та їх призначення.
69. Дайте визначення ітератора. Яке призначення ітераторів у потокових класах?
70. Для чого призначені спеціалізовані шаблони `vector<bool>`, які їх переваги над звичайними масивами об'єктів?
71. Перелічіть основні випадки, де використання лінійних списків є оптимальним.
72. Коли організація даних у двосторонні черги є найбільш оптимальною?
73. Яка відмінність двохсторонніх списків і двосторонніх черг у бібліотеці `TSL`?
74. Чому стек називають адаптером контейнера?
75. Яка відмінність черг і черг з пріоритетами?
76. За рахунок чого асоціативні контейнери забезпечують швидкий доступ до даних?
77. Перелічіть типи асоціативних контейнерів.
78. Яка відмінність словників, словників з дублікатами, множин, множинам з дублікатами і бітових множин?

79. Чим відрізняються словники від словників з дублікатами, множин від множин із дублікатами?
80. Яке призначення бітових множин?
81. Чому використання функціональних об'єктів у бібліотеці TSL є таким поширеним?
82. Перелічіть основні функції немодифікуючих алгоритмів бібліотеки TSL.
83. Перелічіть основні функції модифікуючих алгоритмів бібліотеки TSL.
84. Які алгоритми сортування реалізовані в бібліотеці TSL?
85. Яка відмінність пірамід і множин? Які алгоритми для роботи з пірамідами і множинами реалізовані в бібліотеці TSL?
86. Які засоби підтримки мов реалізовані в стандартній бібліотеці C++?
87. Як необхідно реалізувати роботу програми для коректного виділення та звільнення пам'яті C++?
88. Які засоби підтримки динамічної ідентифікації типів реалізовані в стандартній бібліотеці C++?
89. Які засоби діагностики та локалізації реалізовані в стандартній бібліотеці C++?
90. Назвіть стандартні засоби, реалізовані у C++ для роботи з комплексними числами.

СПИСОК ЛІТЕРАТУРИ

Основна

1. *Бабэ Б.* Просто и ясно о Borland C++. — М.: БИНОМ, 1995. — 400 с.
2. *Бадд Т.* Объектно-ориентированное программирование в действии. — СПб.: Питер, 1997. — 464 с.
3. *Баженова И. Ю.* Visual C++ 6.0. Уроки программирования. — М.: Диалог-МИФИ, 1999. — 416 с.
4. *Брукс Ф.* Мифический человек-месяц, или как создаются программные комплексы. — М.: Символ-Плюс, 2000. — 304 с.
5. *Буч Г.* Объектно-ориентированный анализ и проектирование с примерами на C++. — М.: БИНОМ, 1998. — 560 с.
6. *Вирт Н.* Алгоритмы + структуры данных = программы. — М.: Мир, 1985. — 406 с.
7. *Голуб А. И.* С и C++. Правила программирования. — М.: БИНОМ, 1996. — 272 с.

8. *Дэвис С.* С++ для “чайников”. — К.: Диалектика, 1996. — 304 с.
9. *Карпов Б., Баранова Т.* С++: специальный справочник. — СПб.: Питер, 2000. — 480 с.
10. *Патлас К., Мюррей У.* Visual С++6: Руководство разработчика. — К.: ВНУ, 2000.
11. *Подбельский В. В.* Язык С++. — М.: Финансы и статистика, 1996.
12. *Страуструп Б.* Язык программирования С++. — 3-е изд. — М.: Бином, 1999.

Додаткова

13. *Аммераль Л.* STL для программистов на С++. — М.: ДМК, 1999.
14. *Грегори К.* Использование Visual С++6. — М.: Вильямс, 1999.
15. *Киммел П.* Borland С++5. — СПб.: ВНУ, 1997.
16. *Крейг Арнуш.* Borland С++: освой самостоятельно. — М.: Бином, 1997.
17. *Ласло М.* Вычислительная геометрия и компьютерная графика на С++. — М.: БИНОМ, 1997. — 304 с.
18. *Лейнекер Р.* Энциклопедия Visual С++6. — СПб.: Питер, 1999.
19. *Луис Д.* С и С++: Справочник. — М.: Бином, 1997.
20. *Луис Д.* С и С++: Справочник. — М.: Восточная книжная компания, 1997. — 592 с.
21. *Мейерс С.* Эффективное использование С++. — М.: ДМК, 2000. — 240 с.
22. *Неформальное введение в С++ и Turbo Vision.* — Галерея “Метрополь”, 1992. — 381 с.
23. *Подбельский В. В.* Язык Си++: Учеб. пособие. — М.: Финансы и статистика, 1996. — 560 с.
24. *Пол Айра.* Объектно-ориентированное программирование на С++. — 2-е изд. — М.: Бином, 1999.
25. *Романовская Л. М. и др.* Программирование в среде Си для ПЗВМ ЕС. — М.: Финансы и статистика, 1992. — 352 с.
26. *Секунов Н. Ю.* Самоучитель Visual С++6. — СПб.: ВНУ, 1999.
27. *Скляр В. А.* Язык С++ и ООП. — Минск: Вышэйш. шк., 1997.
28. *Страуструп Б.* Дизайн и эволюция языка С++. — М.: ДМК, 2000. — 448 с.
29. *Страуструп Б.* Язык программирования С++. — СПб.: БИНОМ, 1999. — 991 с.
30. *Фейсон Т.* Объектно-ориентированное программирование на С++ 4.5. — К.: Диалектика, 1996.
31. *Шилдт Г.* Самоучитель С++, — 2-е изд. — СПб.: ВНУ, 1998.
32. *Шилдт Г.* Теория и практика С++. — СПб.: ВГУ, 1996. — 416 с.
33. *Элджер Дж.* С++: библиотека программиста. — СПб.: Питер, 1999.

ЗМІСТ

Пояснювальна записка.....	3
Методичні вказівки до підготовки, написання та захисту реферату.....	10
Індивідуально-консультаційна робота.....	12
Теми і плани самостійної роботи. Структура та форма звітності.....	13
Питання для самоконтролю.....	31
Список літератури.....	34



Відповідальний за випуск *А. Д. Вегеренко*
Редактор *Л. С. Тоболіч*
Комп'ютерне верстання *Н. І. Нечипоренко*

МАУП

Зам. № ВКЦ-3665
Міжрегіональна Академія управління персоналом (МАУП)
03039 Київ-39, вул. Фрометівська, 2, МАУП