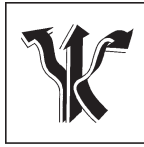
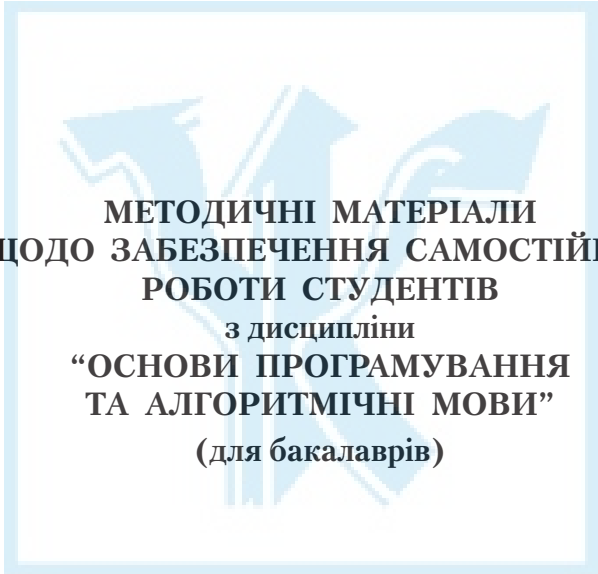


МІЖРЕГІОНАЛЬНА  
АКАДЕМІЯ УПРАВЛІННЯ ПЕРСОНАЛОМ



МАУП



**МЕТОДИЧНІ МАТЕРІАЛИ  
ЩОДО ЗАБЕЗПЕЧЕННЯ САМОСТІЙНОЇ  
РОБОТИ СТУДЕНТІВ  
з дисципліни  
“ОСНОВИ ПРОГРАМУВАННЯ  
ТА АЛГОРИТМІЧНІ МОВИ”  
(для бакалаврів)**

МАУП

Київ 2008

Підготовлено доцентом кафедри прикладної математики та програмування  
*Н. М. Москальковою*

Затверджено на засіданні кафедри інформатики та інформаційних  
технологій (протокол №13 від 30.08.07)

*Схвалено Вченою радою Міжрегіональної Академії управління персоналом*

**Москалькова Н. М.** Методичні матеріали щодо забезпечення самостійної роботи студентів з дисципліни “Основи програмування та алгоритмічні мови” (для бакалаврів). — К.: МАУП, 2008. — 30 с.

Методичні матеріали містять пояснювальну записку, тематику самостійної роботи з дисципліни “Основи програмування та алгоритмічні мови”, (питання для самоконтролю та співбесіди, теми рефератів, тестові завдання), а також список літератури.

Призначена для методичного забезпечення самостійної роботи з дисципліни “Основи програмування та алгоритмічні мови” студентів денної форми навчання, які здобувають освіту за спеціальністю “Програмне забезпечення автоматизованих систем”.

© Міжрегіональна Академія  
управління персоналом (МАУП),  
2008

## **ПОЯСНЮВАЛЬНА ЗАПИСКА**

Сучасна реформа вищої освіти — це насамперед перехід від парадигми навчання до парадигми освіти, самоосвіти. Тому при реформуванні вищої школи, введенні кредитно-модульної технології навчання значно зростає роль самостійної роботи студентів. Самостійна робота студентів є основним засобом опанування навчального матеріалу у позааудиторний час. Студент, який хоче якомога краще оволодіти професією, має добре розуміти: на занятті викладач подає основи знань, навчає, як учити, виділяє ті ключові істини дисципліни, які пробуджують у молодій людині потяг до поглиблення й удосконалення усіх знань. Лише постійне самостійне навчання дає можливість якомога ближче підійти до вершини знань певної галузі, оволодіти такою сумою знань і вмінь, які б дали змогу заявити про себе як про професіонала.

Самостійна робота студентів є надзвичайно важливою складовою підготовки спеціалістів з напрямку “Комп’ютерні науки”, зокрема зі спеціальності “Програмне забезпечення автоматизованих систем”. Теоретичний матеріал з програмування потребує багаторазового підкріплення практичними прикладами. Студенти мають здобути навички самостійного виконання усіх етапів розробки програмного забезпечення (проектування, створення, тестування тощо). Це вимагає від студента систематичного виконання практичних завдань протягом семестру та підготовки до кожного практичного заняття.

Самостійна робота студента повинна бути спланована, організована і методично спрямована як особиста творча праця безпосередньої взаємодії з викладачем. Згідно з державними стандартами навчальний матеріал навчальної дисципліни, передбачений робочим навчальним планом для засвоєння студентом у процесі самостійної роботи, вноситься 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 навчального закладу, навчальних кабінетах, комп’ютерних класах (лабораторіях), а також у домашніх умовах. Навчальний час, відведений для самостійної роботи, регламентується робочим навчальним планом і повинен згідно з Болонською декларацією становити не менше 50% загального обсягу навчального часу студента, відведеного для вивчення конкретної дисципліни. За потреби ця робота проводиться відповідно

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 з використанням складного обладнання чи устаткування, 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 здійснюватись через:

- запам'ятовування певної інформації за рахунок уважного слухання і конспектування лекцій; активної роботи під час практичних занять;
- роботу над конспектами лекцій, планами практичних занять;
- опрацювання літературних джерел (конспектування самостійно вивченого матеріалу, рефератування);
- 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 спрямування студентів у найбільш раціональному напрямі щодо вивчення навчальної дисципліни і акцентуванні уваги на найбільш складних, вузлових питаннях навчальної дисципліни. 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 одночасно засвоюється до 20 % інформації. Викладання інформатики в комп'ютерних класах або в аудиторіях, обладнаних мультимедійним обладнанням (наприклад, мультимедійним проектором або сенсорним екраном), водночас з демонстрацією студентам прийомів роботи з користувальницьким інтерфейсом програми дає змогу 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 або недостатньо розкриті деякі питання практичного заняття, або вони винесені на самостійне опрацювання, студент повинен звернутися до рекомендованих підручників, навчальних посібників і відповідних методичних матеріалів. Підготовку для практичного заняття найкраще здійснювати з використанням ПЕОМ зі встановленим на ньому відповідним програмним забезпеченням. За такої можливості слід використовувати інтерактивні довідкові системи програм *MS Office* та інформаційно-пошукові системи Internet.

*Вивчення навчального матеріалу за підручниками, навчальними посібниками, методичними вказівками, опрацювання матеріалу за першоджерелами, науковою і спеціальною літературою.* Працювати із підручниками, навчальними посібниками, методичними вказівками, практикумами, науковою і спеціальною літературою незалежно від типу їх носія (паперового чи електронного) необхідно так, щоб отримати максимум теоретичних знань і навичок. Під час роботи з цими джерелами студент насамперед повинен ознайомитися з їх змістом, щоб визначити, чи потрібно опрацювати це джерело і чи стосується воно навчального курсу, що вивчається, і тільки після цього визначити послідовність його опрацювання і відібрати необхідний для вивчення матеріал із цього джерела (глави, розділи тощо). У 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 подання навчального матеріалу, а саме: за допомогою гіперпосилань знаходити відповіді на поставлені питання. При опрацюванні матеріалу необхідно з'ясувати сутність питання, що вивчається, не уникаючи при цьому визначення сутності незрозумілих чи незнайомих слів, термінів. Саме інтерактивні гіпертекстові електронні джерела (довідки в складі програмних продуктів, електронні посібники та словники) дають змогу конкретизувати терміни та визначення якнайшвидше. У процесі вивчення матеріалу необхідно аналізувати прочитане, порівнюючи з прослу-

ханою та законспектованою лекцією, робити логічні висновки, позначати незрозумілі положення з метою їх подальшого з'ясування на практичному занятті. Бажано відпрацювати зручну для себе певну 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 (насамперед *MS Word*) надають можливість створення електронного конспекту з примітками, виносками, коментарями та його роздруківки. Для самостійного поглибленого вивчення навчального матеріалу студенту слід звертатися до наукової та спеціальної літератури, яка може і не бути зазначена в навчально-методичному комплексі. Використання самостійно отриманих відомостей як у навчанні, так і на практиці є, безперечно, цінним здобутком діяльності студента на шляху формування свого професійного потенціалу.

*Робота з бібліотечними фондами та дистанційними джерелами з метою пошуку необхідної інформації.* Знання з інформатики належать до базової підготовки сучасної людини. Вони становлять основу для подальшого засвоєння спеціалізованого програмного забезпечення за фаховою освітою і після закінчення навчального закладу застосовуватимуться в будь-якому виді діяльності. З позицій випереджаючої освіти навчання тільки за конспектом лекцій і основною літературою, зазначеною у навчальній програмі, є недостатнім. У більшості випадків належна підготовка вимагає вмінь швидко знаходити та опрацьовувати 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 та базах знань, доступ до яких здійснюється за допомогою відповідних сервісів *Internet* (переважно *Word Wide Web*, *FTP* та *UseNet newsgroups*).

Для пошуку документа використовуються різні його ознаки. Наприклад це — реквізити документа (УДК. Автор(и). Заголовок опису. Основний заголовок: відомості, що належать до заголовка/Відомості про відповідальність. — Відомості про видання (у тому числі URL-адреса Web-документа або Ftp-файла). — Місце видання, дата видання. — Обсяг). УДК — це універсальна десяткова класифікація

будь-яких офіційних видань у всьому світі. Відповідні довідники видаються багатьма мовами і постійно оновлюються. В Україні у 2006 р. Книжковою палатою України імені Івана Федорова видано “Універсальну десяткову класифікацію. Зміни та доповнення. Випуск 4” в паперовому варіанті. Довідкова база УДК постійно нарощується за рахунок електронних видань. Знання УДК дає змогу швидко знайти необхідне джерело за систематичним бібліотечним каталогом. Наприклад, УДК видань з інформаційних технологій починається з 004.

Якщо код УДК невідомий, 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 питань з 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 утворюють 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 запиту слід ввести ключове слово (поєднання слів), клацнути **Пошук**, і система повідомить, чи є відповідний розділ в її каталозі і запропонує в нього перейти, минувши 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 мережі *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 зводиться до того, що користувач вводить в спеціальному полі (або в кількох полях) ключові слова та/або словосполучення, що найточніше відображають сутність проблеми.

До загальних положень мов запитів належать:

- Ключові слова можна вводити у відповідне поле пошукової системи поодиночі, послідовно звужуючи пошук, або ж вводити відразу кілька слів, розділяючи їх пробілами або комами. Регістр не має значення.
- Режим пошуку “AND” (“І”) означає, що будуть знайдені тільки ті дані, де зустрічається кожне з ключових слів.
- При використанні режиму “OR” (“АБО”) результатом пошуку будуть всі дані, де зустрічається хоча б одне ключове слово.
- 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 яке-небудь слово з пошуку, поставте перед ним знак “-”. Наприклад: “+таблиці -Excel”.
- 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 ключо-



вого слова означає, що будуть знайдені тільки слова точно відповідні запиту (наприклад, “текстовий! редактор!”).

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

- Якщо для пошуку потрібно ввести словосполучення, укладіть його в лапки.
- Якщо ви пишете все слово рядковими буквами, будуть знайдені 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 приніс результатів, можливо, при введенні ви припустилися помилки.
- 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 ближчий до шуканої теми, ніж інші, клацніть **Знайти схожі документи**. Це посилання розташоване під короткими описами знайдених документів. Система проаналізує сторінку і знайде документи, схожі на той, що ви вказали.

Подібних систем в *Internet* значно більше, ніж тематичних каталогів. Серед пошукових систем існують як обширні з тематики метапошукові системи, так і вузькоспеціалізовані. Найбільш відомі з них: <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](http://uk.wikipedia.org), [www.bukinist.agava.ru](http://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://www.zodchiy.ru/links/search/>.

[websearch.report.ru/](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 або паперових носіях або 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.

**Метою** вивчення дисципліни “ Основи програмування і алгоритмічні мови” є опанування студентами знаннями, уміннями і навичками з основ програмування і кодування алгоритмів алгоритмічними мовами Паскаль та С.

У курсі досліджуються питання, пов’язані з процедурними мовами програмування високого рівня. Розглядаються базові концепції програмування, вивчаються основні алгоритмічні конструкції: прості та складені оператори, умовні оператори, цикли, рекурсія. Знайомство із структурами даних передбачає роботу як з даними статичної структури (прості дані, масиви, множини, записи), так і з даними динамічної структури (файли, лінійні однозв’язні та двозв’язні списки, стеки, черги). Значна увага приділяється набуттю корисних технічних навичок, а саме — роботі із зовнішніми файлами, створенню звукових та графічних ефектів.

## **ТЕМАТИКА САМОСТІЙНОЇ РОБОТИ**

### **Тема 1. Знайомство з інтегрованим середовищем Turbo Pascal 7.0**

1. Принципи структурного та модульного програмування.
2. Загальна структура паскалівської програми. Написання програм з лінійною структурою.

3. Основні прийоми роботи в середовищі Turbo Pascal. Прогін та відлагодження програми. Клавіатурні комбінації при роботі в текстовому редакторі. Довідкова служба середовища Turbo Pascal.

*Література* [4; 7; 10; 13; 14; 16–18]

### **Теми рефератів**

1. Редагування тексту програми в середовищі Turbo Pascal.
2. Написати мовою Паскаль програму знаходження різниці тризначним числом та числом, що складається з тих самих цифр, що записані в оберненому порядку.
3. Написати мовою Паскаль програму знаходження суми цифр чотиризначного числа.
4. Написати мовою Паскаль програму обчислення довжини кола, площі кола та об'єм шару одного заданого радіуса.
5. Написати мовою Паскаль програму обчислення площі та периметра трикутника, заданого координатами трьох вершин.

### **Питання для самоконтролю та співбесіди**

1. Схарактеризуйте поняття структурного програмування.
2. Що таке модульне програмування?
3. Що означає проектування згори вниз та знизу вгору?
4. З яких основних частин складається паскалівська програма? Яке розширення мають початкові тексти паскалівських програм?
5. Які клавіатурні комбінації для редагування текстів паскалівських програм вам відомі?

### **Тема 2. Основні елементи мови Pascal**

1. Алфавіт. Ідентифікатори. Константи. Вирази. Класифікація та пріоритет операцій. Коментарі.
2. Структура програми. Прості (оператор присвоювання, оператор процедури, оператор переходу) та складені оператори.
3. Умовні оператори (умовний оператор if, оператор вибору варіанту case).
4. Програмування циклів. Оператори циклу з передумовою та постумовою. Оператор циклу з лічильником.

*Література* [4; 7; 10; 12; 13; 14; 16–18]

## Теми рефератів

1. Написати мовою Паскаль програму, після виконання якої менше з двох нерівних чисел буде збільшене вдвічі, а більше — залишене без змін.
2. Написати мовою Паскаль програму для розв'язання квадратного рівняння  $ax^2 + bx + c = 0$ .
3. Написати мовою Паскаль програму для розв'язання бікватратного рівняння  $ax^4 + bx^2 + c = 0$ .
4. Написати мовою Паскаль програму визначення цифр цілого числа  $n$ .
5. Написати мовою Паскаль програму визначення всіх натуральних чисел, які менші за  $n$ , квадрат суми цифр яких дорівнює заданому числу  $m$ .

## Питання для самоконтролю та співбесіди

1. У чому полягає різниця між змінними та константами?
2. Наведіть приклади, коли доцільно використовувати оператор вибору.
3. Наведіть приклади, коли доцільно використовувати оператор циклу з передумовою.
4. Наведіть приклади, коли доцільно використовувати оператор циклу з постумовою.
5. Наведіть приклади, коли доцільно використовувати оператор циклу з лічильником.

## Тестові завдання

### 1. Рівність двох значень визначає такий оператор:

- а) = ;
- б) == ;
- в) := ;
- г) =. .

### 2. Синтаксична помилка допущена в такому умовному операторі:

- а) if B = 0 then Writeln('Деление на нуль невозможно.');
- б) if a < b then max := a else max := b;
- в) if (a>b) and (b>0) then c:=a+b;
- г) if a < b then min := a; else min := b.

### 3. При компіляції програми

**Program** Example;

```
var
  a, b: integer;
begin
  readln(a);
  b := a + 17; c := b div 3;
  writeln(c)
end.
end.
```

**виникла така помилка:**

- a) “;” expected;
- б) Unknown identifier;
- в) Unexpected end of file;
- г) Invalid floating point operation.

### 4. При компіляції програми

```
program Test;
Var a,b:integer;
begin
  a := -100;
  b := sqrt(a);
  writeln(a);
end.
```

**виникла така помилка:**

- a) “;” expected;
- б) Unknown identifier;
- в) Type mismatch;
- г) Invalid floating point operation.

### 5. Результатом роботи такої програми є:

**Program** Transpose;

```
var
  n, n1, p, a, i : longint;
begin
  write('Введите натуральное число n'); readln(n).
  a := n; i := 1;
  p := n mod 10;
  repeat
    i := i*10; n := n div 10
```

```

until n<10;
n1 := a - n*i - p + n + p*i;
writeln('Число после перестановки', n1)
end.

```

- а) число, що отримане перестановкою усіх цифр заданого числа;
- б) число, що отримане перестановкою першої та останньої цифр заданого числа;
- в) число, що є добутком цифр заданого числа;
- г) число, що є сумою цифр заданого числа.

### **Тема 3–4. Прості та структуровані типи даних**

1. Порядкові типи даних. Група цілих типів (Shortint, Integer, Longint, Byte, Word). Група дійсних типів (Single, Real, Double, Extended, Comp).
2. Дані логічного типу (Boolean). Символьний тип даних (Char).
3. Типи даних користувача. Перелічувані типи даних. Інтервальні типи даних.
4. Масиви та основні операції над ними. Рядки.
5. Комбіновані типи даних (записи). Записи з варіантами. Робота із записами.
6. Множини. Операції над множинами.

*Література* [4; 7; 10; 13; 14; 16–18]

### **Тема рефератів**

1. Написати мовою Паскаль програму знаходження для заданої матриці  $A(N,N)$ 
  - суми всіх елементів;
  - суму елементів головної діагоналі;
  - значення найбільшого та найменшого елементів головної діагоналі.
2. Дано дві послідовності по 30 цілих чисел в кожній. Написати мовою Паскаль програму знаходження найменшого числа з тих чисел першої послідовності, які не входять до другої послідовності.
3. Написати мовою Паскаль програму циклічного зсуву елементів масиву на одну позицію вліво:  $x\_new[n]=x[1]$ ,  $x\_new[k]=x[k+1]$  при  $k=1, 2, \dots, n-1$ .
4. Нехай

```

Type info = record
    Imja:string;

```

Pol: (m,f);  
Rost:140..200

End;

group=array [name] of info;

Написати мовою Паскаль програму визначення

- а) середнього зросту жінок;
- б) ім'я найвищого чоловіка;
- в) факту наявності в групі людей одного зросту.

5. Написати мовою Паскаль програму визначення того, чи є послідовність чисел монотонно зростаючою.

### Питання для самоконтролю та співбесіди

1. Дайте порівняльну характеристику цілих та дійсних типів.
2. Дайте визначення масиву як структури даних.
3. Стандартні процедури і функції для виконання операцій над рядками.
4. Як здійснюється звернення до полів запису? Наведіть приклади з поясненнями.
5. Опишіть відомі методи сортування масивів.

### Тестові завдання

#### 1. Тип *Word* належить до:

- а) цілочисельних типів даних;
- б) рядкових типів даних;
- в) символьних типів даних;
- г) типу даних, який може бути визначений користувачем.

#### 2. Правильним є такий опис інтервального типу:

- а) Type n = -100 to 100;
- б) Type n = -100 ... 100;
- в) Type n : -100 to 100;
- г) Type n : -100 ... 100;

#### 3. Помилковими є такі описи:

- а) Operator: (plus, minus, times);
- б) Matrix: array[1.10,1.10] of Real;
- в) Done,Error: boolean;
- г) I, J, K: integer.

#### **4. Помилковими є такі описи:**

- a) a: set of boolean;
- б) b: set of false..true;
- в) c: set of integer;
- г) d: set of string.

#### **5. З наведених помилковими є такі описи, якщо color: (red, green, blue):**

- a) color:=succ(red);
- б) color:=pred(green);
- в) color:=succ(blue);
- г) write(color).

#### **Тема 5. Програмування і використання процедур і функцій**

- 1. Опис процедур і функцій. Структура процедур і функцій.
- 2. Область дії ідентифікаторів при використанні процедур і функцій.
- 3. Параметри-значення та параметри-змінні. Особливості передачі параметрів.

*Література* [4; 7; 10; 12; 13; 14; 16–18]

#### **Теми рефератів**

- 1. Написати мовою Паскаль логічну функцію `sravnenie` (a, b), яка порівнює два раціональних числа a та b, які задані за допомогою типу

```
type
  rational=record
    chisl: integer;
    znam: 1..maxint
  end;
```

- 2. Написати мовою Паскаль процедуру `add(a, b, c)`, яка додає раціональні числа a і b та додає їх суму раціональному параметру c/.

- 3. Написати мовою Паскаль рекурсивну функцію `C(m,n)`, де  $0 \leq m \leq n$  для обчислення біноміального коефіцієнта  $C_n^m$  за формулою

$$C_n^0 = C_n^n = 1; C_n^m = C_{n-1}^m + C_{n-1}^{m-1} \text{ при } 0 < m < n$$



4. Написати мовою Паскаль рекурсивну процедуру виведення на екран цифр цілого натурального числа в оберненому порядку.

5. Написати мовою Паскаль рекурсивну функцію обчислення чисел Фібоначчі.

### **Питання для самоконтролю та співбесіди**

1. Чим відрізняються процедура та функція?
2. Як використовуються формальні та фактичні параметри?
3. Що таке локальні та глобальні параметри?
4. Як визначити область дії ідентифікаторів?
5. Які існують особливості використання рекурсивних процедур та функцій?

### **Тема 6–7. Робота із зовнішніми файлами.**

#### ***Робота з динамічними структурами даних***

1. Класифікація файлів. Доступ до файлів. Процедури і функції для роботи з файлами. Процедури і функції для роботи з типізованими файлами. Текстові файли.
2. Динамічні структури даних. Виділення та звільнення пам'яті. Динамічна пам'ять. Вказівники.
3. Зв'язані динамічні дані. Лінійні однозв'язні списки. Робота з чергою. Робота із стеком.
4. Обробка лінійних двозв'язних списків.

*Література* [4; 7; 10; 12; 13; 14; 16–18]

### **Теми рефератів**

1. Дано файл, компоненти якого є цілими числами. Вивести на екран максимальне число з перших 10 чисел, потім — других 10 и так далі.
2. Дано файл, компоненти якого є цілими числами. Створити два файли, в першому розмістити усі парні числа, у другому — всі непарні.

3. Сформувати файл записів

```
Type Klass=record  
    Imja:string;  
    telefon: 10000..99999;  
    adres:string  
End;
```

Написати мовою Паскаль програму визначення:

- телефону заданої особи;
- особи, якій належить телефон;
- списку людей, що живуть на одній вулиці.

### **Питання для самоконтролю та співбесіди**

1. Як здійснити додавання елемента в кінець однозв'язного та двозв'язного списку?
2. Як здійснити додавання елемента на початок однозв'язного та двозв'язного списку?
3. Як здійснити вставку елемента в середину однозв'язного та двозв'язного списку?
4. Як здійснити вилучення елемента з однозв'язного та двозв'язного списку?
5. Як здійснити виведення на екран елементів однозв'язного та двозв'язного списку?

### **Тема 8. Програмування графіки. Обробка переривань від клавіатури. Звукові ефекти**

1. Організація роботи із зовнішніми пристроями в середовищі Turbo Pascal.
2. Графічні можливості середовища Turbo Pascal.
3. Звукові ефекти. Процедури Sound, Delay та NoSound модуля CRT.

*Література [4; 7; 10; 12–14; 16–18]*

### **Теми рефератів**

1. Написати мовою Паскаль програму відображення на екрані графіка функції, значення якої обчислюється за допомогою функції Func(x).
2. Написати мовою Паскаль програму малювання різнокольорових концентричних кіл.
3. Написати мовою Паскаль програму, яка відображає на екрані переміщення шарика по синусоїді.
4. Написати мовою Паскаль програму управління переміщенням об'єкта.
5. Написати мовою Паскаль програму наближеного обчислення інтегралу функції та графічно відобразити результати обчислення.

### Питання для самоконтролю та співбесіди

1. Чим відрізняються текстовий та графічний режим роботи?
2. Назвіть особливості розмітки екрана у графічному режимі.
3. Які процедури можна використовувати для керування кольором у графічному режимі?
4. Які процедури використовуються для керування роботою у графічному режимі?
5. Яким чином здійснити обробку переривань від клавіатури?

### Теми 9–10. Основні характеристики мови С.

#### Типи, оператори та вирази мови

1. Структура програми, літерали, синтаксис мови, зовнішні змінні, область їх дії.
2. Типи даних, їх розміри.
3. Арифметичні та логічні оператори, оператори відношення та унарні оператори. Перетворення типів. Оператори присвоювання. Умовні вирази.

*Література* [1; 3; 5; 6; 8; 9; 11; 19]

#### Теми рефератів

1. Написати мовою С програму визначення обсягу та площі бокової поверхні циліндра із заданим радіусом та висотою.
2. Написати мовою С програму віднімання від більшого числа меншого.
3. Написати мовою С програму піднесення числа  $a$  до ступеня  $b$ .
4. Написати мовою С програму обчислення значень  $a$  та  $b$  за формулами

$$a = \frac{2\sqrt{\sin^2 x - 1,5 \cdot 10^{-3}}}{\frac{1}{3} + e^{-2x}}; \quad b = x - \frac{2^2}{5} + \frac{1}{3} + e^{-2z}.$$

5. Написати мовою С програму обчислення добутку цифр заданого чотиризначного числа.

### Питання для самоконтролю та співбесіди

1. Що буде результатом виконання операції  $x | y \& z$ , якщо  $x = 03$ ;  $y = 02$ ;  $z = 01$ ?
2. Яку структуру має програма мовою С?

3. Які прості типи даних використовуються в програмах мовою C?
4. Як здійснюється перетворення типів у програмах мовою C?
5. У чому полягає різниця між умовною операцією та умовним оператором у мові C?

### Тестові завдання

**1. Змінна `int x` після виконання оператора `x=(int)3.8+3.3` матиме таке значення:**

- a) `x=6;`
- б) `x=7.1;`
- в) `x=7;`
- г) `x=3.`

**2. У результаті виконання програми**

```
#include <stdio.h>
int main()
{
    char a,b;
    a=126;
    b=a+127;
    printf("%d", b);
}
```

**буде отримано такий результат:**

- a) 253;
- б) -3;
- в) 126;
- г) 127.

**3. У результаті виконання програми**

```
#include <stdio.h>
int main()
{
    char a;
    a=130;
    printf("%d %c ",a, a);
}
```

**буде отримано такий результат:**

- a) -126 -126;
- б) -126;
- в) 130;
- г) 130 130.

#### 4. У результаті виконання програми

```
#include <stdio.h>
int main()
{
    int a,c;
    float b=1;
    a=2;
    c=3;
    b=a+c+5.5;
    printf(“%d %2.3f %d”, a,b,c);
}
```

**буде отримано такий результат:**

- a) 8.5 8.500 8.5;
- б) 8.5 8.500 3;
- в) 8 8.000 3;
- г) 8 8.500 3.

#### 5. У результаті виконання програми

```
#include <stdio.h>
int main()
{
    int a=2;
    const float b=1;
    if (b=a+0.1) {printf(“%2.2f”,b);}
    else printf(“%d”, a);
}
```

**буде отримано такий результат:**

- a) помилка при компіляції
- б) 2.10;
- в) 2;
- г) 2.00;
- д) 1.00.

#### **Теми 11–12. Управління послідовністю виконання обчислень. Функції, блокова структура**

1. Умовні інструкції. Багатоступеневі конструкції. Перемикачі. Цикли, цикли з передумовою та постумовою. Інструкції переривання циклів. Позначки та інструкції переходу на позначку.
2. Використання функцій. Внутрішні та зовнішні змінні.
3. Особливості створення багатофайлових програм.
4. Використання Сі-препроцесора.

*Література* [1; 3; 5; 6; 8; 9; 11; 19]

## Теми рефератів

1. Написати мовою С програму обчислення виразу значення  $y = \frac{(n! + m!)}{(n + m)!}$ .
2. Написати мовою С програму знаходження кореня рівняння  $f(x) = 0$  на відрізку  $[a, b]$  ( $f(a)f(b) < 0$ ,  $f$  — неперервна і монотонна на  $[a, b]$ ).
3. Написати мовою С рекурсивну функцію знаходження  $x^n$ .
4. Написати мовою С програму визначення першого елемента геометричної прогресії, якщо задано  $n$ -й елемент та її знаменник.
5. Написати мовою С програму визначення того, чи є послідовність чисел монотонно спадаючою.

## Питання для самоконтролю та співбесіди

1. Опишіть механізм роботи циклу з лічильником у мові С.
2. Опишіть механізм роботи циклу з передумовою у мові С.
3. Опишіть механізм роботи циклу з постумовою у мові С.
4. Для чого призначені оператори continue і break у мові С?
5. Як застосовуються макропідстановки у мові С?

## Тестові завдання

### 1. У результаті виконання програми

```
#include <stdio.h>
int main()
{
    int a,b;
    a=1;
    b=2;
    if (b--==++a) {printf(“%d + %d = %d”,a,b,a+b);}
    else {printf(“%d - %d = %d”,a,b,a-b);}
}
```

**буде отримано такий результат:**

- a)  $2 + 1 = 3$ ;
- б)  $2 + 2 = 4$ ;
- в)  $2 - 1 = 1$ ;
- г)  $2 - 2 = 0$ .

**2. У результаті виконання програми:**

```
#include <stdio.h>
int main()
{
    int a,b;
    a=200;
    b=500;
    if ((a>0)||(b=a-10)) {printf("умова виконана: a=%d
b=%d",a,b);}
    else {printf("умова не виконана: a=%d b=%d",a,b);}
}
```

**буде отримано такий результат:**

- а) умова виконана: a=200 b=500;
- б) умова виконана: a=200 b=190;
- в) умова не виконана: a=200 b=500;
- г) умова не виконана: a=200 b=190.

**3. У результаті виконання програми**

```
#include <stdio.h>
int main()
{
    int a,b;
    a=200;
    b=500;
    if ((a>0)&&(b=a-10)) {printf("умова виконана: a=%d
b=%d",a,b);}
    else {printf("умова не виконана: a=%d b=%d",a,b);}
}
```

**буде отримано такий результат:**

- а) умова виконана: a=200 b=500;
- б) умова виконана: a=200 b=190;
- в) умова не виконана: a=200 b=500;
- г) умова не виконана: a=200 b=190.

**4. У результаті виконання програми**

```
#include <stdio.h>
int main()
{
    int a=0;
    char c='a';
```

```

switch (c)
{
    case 97:a++;
    case 98:a++;
    case 99:a++;
};
printf(" %d ",a);
}

```

**буде отримано такий результат:**

- а) 0;
- б) 1;
- в) 2;
- г) 3.

**5. У результаті виконання програми**

```

#include <stdio.h>
int main()
{
    int a=2;
    float b=1;
    if (b=a+0.1) {printf("%.2f",b);}
    else printf("%d", a);
}

```

**буде отримано такий результат:**

- а) помилка при компіляції
- б) 2.10;
- в) 2;
- г) 2.00;
- д) 1.00.

**Тема 13–14. Показчики, масиви. Структури, об'єднання**

1. Адресація пам'яті. Оператори адресації та розкриття посилання. Зв'язок між масивами та показниками. Ініціалізація масивів.
2. Адресна арифметика.
3. Опис структури, позначення компонентів структур, ініціалізація структур. Показники на структури. Масиви структур. Визначення нових типів. Об'єднання.

*Література [1; 3; 5; 6; 8; 9; 11; 19]*



## Теми рефератів

1. Написати мовою С функцію обчислення добутку двох матриць.
2. Написати мовою С функцію обчислення суми та добутку парних її елементів.
3. Написати мовою С програму, яка міняє місцями найбільший та найменший її елементи.
4. Написати мовою С програму обчислення суми та добутку непарних від'ємних елементів матриці, для яких виконується умова  $|a_{ij}| < i$ .
5. Написати мовою С програму перетворення матриці А за правилом: рядок з номером Р стає стовпцем з номером Р, а стовпець з номером Р стає рядком з номером Р.

### Питання для самоконтролю та співбесіди

1. Чи є показчик змінною? Якого типу?
2. Як одержати дані, на які вказує показчик?
3. Що буде результатом виконання оператора ++\*ip, якщо його описано так: int \*ip?
4. Яким чином в С функції можна змінити значення змінних, що знаходяться у тому блоці, з якого ця функція була викликана?
5. Яким чином в С можна отримати доступ до елементів масиву за допомогою вказівника?

### Тема 15. Введення, виведення

1. Стандартне введення-виведення. Форматне виведення, введення.
2. Доступ до зовнішніх файлів. Введення-виведення символьних рядків.

## Теми рефератів

1. Написати мовою С програму дописування інформації одного файла в кінець іншого.
2. Написати мовою С програму вилучення коментарів з файла.
3. Файл містить інформацію про студентів (прізвище, ім'я, по батькові, дата народження, група, інститут, дата вступу, дата закінчення). Написати мовою С програму вилучення записів про студентів, які закінчили навчання.

4. Написати мовою С програму порівняння двох файлів та виведення на екран першого рядка, в якому файли відрізняються.
5. Написати мовою С програму сортування рядків файлу за довжиною та виведення результату в окремий файл.

### Питання для самоконтролю та співбесіди

1. Назвіть функції С, які використовуються для введення зі стандартного потоку даних.
2. Назвіть функції С, які використовуються для виведення у стандартний потік даних.
3. Які функції управління доступом до файлів використовуються в С?
4. У чому полягають особливості введення-виведення рядків?
5. Що означає прямий доступ до даних файла?

### Тестові завдання

**1. Після виконання оператора `printf("%d", 100+65)` на екран буде виведено:**

- a) 100+65;
- б) 165;
- в) 65+100;
- г) 100.

**2. Після виконання таких операторів**

```
int main()
{
    int n;
    float m;
    n=2;
    m=5.5;
    printf("%f+%d=%f", n, m, n+m);
}
```

**на екран буде виведено:**

- a) 0.000000 + 16406 = 7.500000;
- б) 2.000000 + 16406 = 7.500000;
- в) 2.000000 + 5 = 7.500000;
- г) 2.000000 + 5.5 = 7.500000.

### 3. У результаті виконання програми

```
#include <stdio.h>
int main()
{
    int a=025;
    float b=0xA1F;
    if (b=a+0.1) {printf(“%2.2f”,b);}
    else printf(“%d”, a);
}
```

**буде отримано:**

- a) 0xA1F;
- б) 2591;
- в) 025;
- г) 21.10.

### **Тема 16–18. Аналіз алгоритмів програм. Алгоритми обробки деревоподібних структур. Орієнтовані та неорієнтовані графи**

1. Ємнісна складність алгоритмів. Часова складність алгоритмів. Алгоритми логарифмічної, лінійної, поліноміальної та експоненційної складності.
2. Впорядковані та невлпорядковані дерева. Реалізація дерев за допомогою масивів. Представлення дерев за допомогою списків синів.
3. Орієнтований граф. Основні визначення. Види графів та операції над графами.
4. Представлення орієнтованих графів. Пошук найкоротшого шляху.
5. Обхід неорієнтованих графів. Пошук в глибину. Пошук в ширину.

*Література* [1; 3; 5; 6; 8; 9; 11; 15; 19]

### **Теми рефератів**

1. Дослідження складності алгоритмів впорядкування масивів.
2. Написати мовою С програму, яка дає змогу визначити, чи є граф ациклічним.
3. Написати мовою С програму, яка підраховує кількість листів у бінарному дереві.

4. Написати мовою С програму, яка реалізує пошук у глибину для графів, що представлені списками суміжних вершин.
5. Написати мовою С програму, яка реалізує пошук у ширину для графів, що представлені списками суміжних вершин.

### **Питання для самоконтролю та співбесіди**

1. Як здійснюється прямий, зворотний та симетричний обходи дерев?
2. Схарактеризуйте способи представлення графів.
3. Які існують алгоритми пошуку найкоротшого шляху?
4. Як здійснюється пошук у ширину?
5. Як здійснюється пошук у глибину?

### **СПИСОК ЛІТЕРАТУРИ**

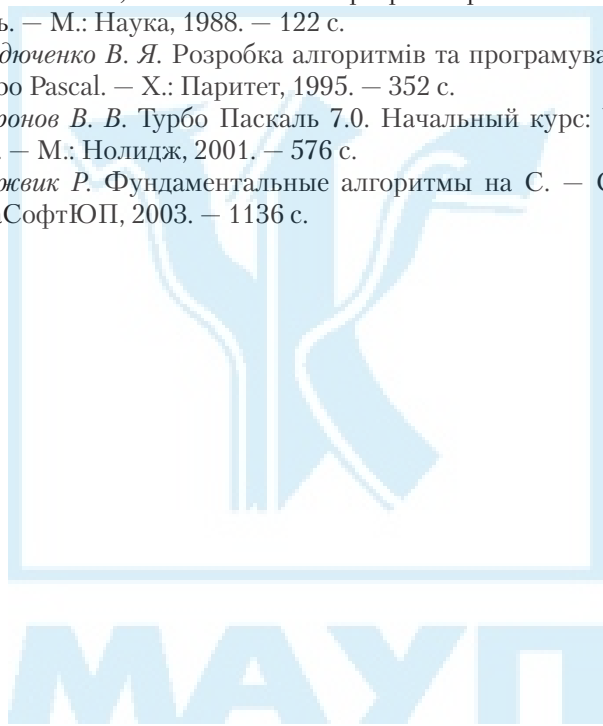
#### *Основна*

1. *Дейтел Х. М., Дейтел П. Дж.* Как программировать на С. — М.: Бином, 2002. — 1156 с.
2. *Карпенко С. Г., Шпортюк Г. А.* Практикум з програмування. Мова С. — К.: МАУП, 2006. — 150 с.
3. *Керниган Б., Ритчи Д.* Язык программирования Си. — М.: Финансы и статистика, 1992. — 272 с.
4. *Ставровский А.* Турбо Паскаль 7.0: Учебник. — К.: BHV, 2000. — 400 с.
5. *Стивен Прата.* Язык программирования С. — М.: DiaSoft, 2002. — 883 с.

#### *Додаткова*

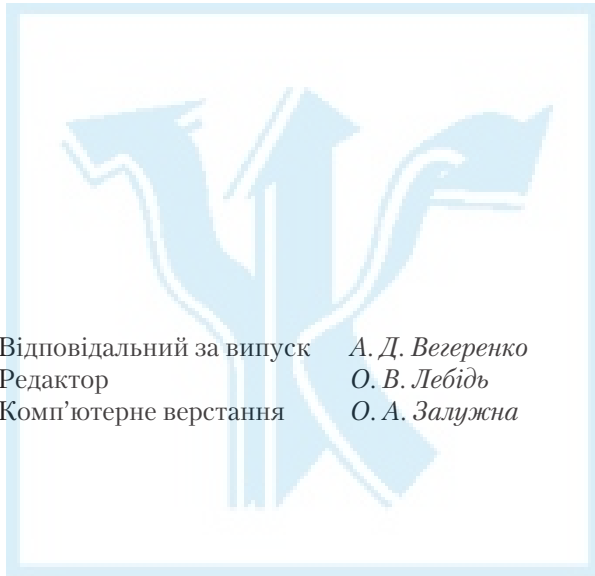
6. *Архангельский А. Я.* Программирование в С++ Builder 6. — М.: Бином, 2004. — 1152 с.
7. *Ахо Хопкрофт Ульман.* Структуры данных и алгоритмы. — М.: Вильямс, 2000. — 384 с.
8. *Белецкий Я.* Энциклопедия языка Си. — М.: Мир, 1992. — 687 с.
9. *Болски М. И.* Язык программирования Си. — М.: Радио и связь, 1988. — 96 с.
10. *Вирт Н.* Алгоритмы + структуры данных = программы. — М.: Мир, 1985. — 406 с.
11. *Герберт Шилдт.* Полный справочник по С. — М.: Вильямс, 2004. — 700 с.

12. *Грогоно П.* Программирование на языке Паскаль. — М.: Мир, 1982. — 384 с.
13. *Кнут Д.* Искусство программирования для ЭВМ. Основные алгоритмы. — М.: Мир, 1976. — 735 с.
14. *Марченко А. И., Марченко Л. А.* Программирование в среде Turbo Pascal 7.0. — К.: ВЕК+Ю, 2000. — 464 с.
15. *Новиков Ф. А.* Дискретная математика для программистов. — СПб.: Питер, 2001. — 304 с.
16. *Семашко Г. Л., Салтыков А. И.* Программирование на языке Паскаль. — М.: Наука, 1988. — 122 с.
17. *Сердюченко В. Я.* Розробка алгоритмів та програмування мовою Turbo Pascal. — Х.: Паритет, 1995. — 352 с.
18. *Фаронов В. В.* Турбо Паскаль 7.0. Начальный курс: Учеб. пособие. — М.: Нолидж, 2001. — 576 с.
19. *Седжвик Р.* Фундаментальные алгоритмы на С. — СПб.: ООО ДиаСофтЮП, 2003. — 1136 с.



## ***ЗМІСТ***

Пояснювальна записка .....	3
Тематика самостійної роботи .....	10
Список літератури.....	28



Відповідальний за випуск	<i>А. Д. Вегеренко</i>
Редактор	<i>О. В. Лебідь</i>
Комп'ютерне верстання	<i>О. А. Залужна</i>

**МАУП**

Зам. № ВКЦ-3364

Міжрегіональна Академія управління персоналом (МАУП)  
03039 Київ-39, вул. Фрометівська, 2, МАУП