**УПРАВЛІННЯ ОСВІТИ І НАУКИ**
**ВОЛИНСЬКОЇ ОБЛАСНОЇ ДЕРЖАВНОЇ АДМІНІСТРАЦІЇ**

**ЛУЦЬКИЙ ПЕДАГОГІЧНИЙ КОЛЕДЖ**

***Циклова комісія викладачів фізико-математичних дисциплін***

**Навчально-методичне забезпечення**

**дисципліни**

# «Програмування з методикою навчання»

# (І семестр)

***Навчально-методичний посібник***

**Луцьк – 2015**

**УДК 004.41**
**ББК 32.973.2-018**
**О-75**

**Навчально-методичне забезпечення дисципліни «Програмування з методикою навчання» (І семестр) :** Навчально-методичний посібник / [Укл. Т. Г. Четверикова] – Луцьк, 2015. – 72 с.

***Автор-укладач: Четверикова Т. Г.***

Розглянуто на засіданні циклової комісії викладачів фізико-математичних дисциплін і рекомендовано до друку (Протокол №1 від 28.08.2015 р.)

Друкується за рішенням науково-методичної ради Луцького педагогічного коледжу (Протокол №1 від 15.09.2015 р.)

*Рецензенти:*

***Гайдай С. І.***, кандидат фізико-математичних наук, доцент кафедри прикладної математики та інформатики Східноєвропейського національного університету імені Лесі Українки

***Собчук О. М.***, кандидат педагогічних наук, доцент кафедри прикладної математики та інформатики Східноєвропейського національного університету імені Лесі Українки

Навчально-методичний посібник містить методичні матеріали з дисципліни «Програмування з методикою навчання»: тексти лекцій, розробки практичних занять, матеріали для організації самостійної позааудиторної та індивідуальної роботи студентів, перелік питань для підготовки до модульних контрольних робіт.

Матеріали, подані у посібнику, стануть у нагоді студентам педагогічних коледжів, які здобувають освіту за спеціальністю 5.01010201 «Початкова освіта» з додатковою кваліфікацією «Вчитель інформатики в початковій школі».

# Передмова

Формування алгоритмічного мислення та вмінь використовувати програмне забезпечення для створення нових програмних продуктів, розв’язання задач користувача – складова інформаційної культури сучасного студента.

Метою вивчення дисципліни «Програмування з методикою навчання» є ознайомлення студентів з основними принципами технології програмування та оволодіння основами мов програмування, методами проектування та створення програм згідно сучасних технологій програмування, формування фундаментальної теоретичної бази, вироблення практичних навичок свідомого і раціонального використання комп'ютерів у повсякденній практичній діяльності, формування та розвиток просторового мислення, що є одним з важливих показників інтелектуального розвитку.

Основними завданнями вивчення дисципліни "Програмування з методикою навчання" є:

- ознайомлення з ідеологією розробки програм засобами програмування;
- ознайомлення з середовищем розробки програм засобами програмування;
- сполучення традиційних навичок програмування з новими засобами;
- опанування основних засобів створення програм засобами візуального об’єктного програмування.

В результаті вивчення курсу студенти повинні набути базові знання, на основі котрих у межах дисциплін педагогічного та психологічного циклів, що вивчаються у наступних семестрах, будуть формуватися такі вміння й навички роботи з інформацією за допомогою комп’ютера й інформаційно-комунікаційних технологій (ІКТ), які дозволяють у подальшому всебічно, усвідомлено й ефективно використовувати комп’ютер і засоби ІКТ у своїй професійній діяльності.

Пропонований навчально-методичний посібник містить матеріали лекцій, розробки практичних занять, матеріали для організації самостійної позааудиторної роботи студентів, перелік питань для підготовки до модульних контрольних робіт та семестрового екзамену з дисципліни «Програмування з методикою навчання». Матеріали, подані у посібнику, стануть у нагоді студентам педагогічних коледжів, які здобувають освіту за спеціальністю 5.01010201 «Початкова освіта» з додатковою кваліфікацією «Вчитель інформатики в початковій школі».

# Зміст

## Тематичний план з дисципліни «Програмування з методикою навчання» (І семестр)

| *№ заняття* | *Тема заняття* | *К-ть год.* | *Тип заняття* |
|---|---|---|---|
| **Змістовий модуль 1. Системи числення. Математичні основи роботи комп'ютера** | | | |
| 1<br>2 | Поняття про системи числення. Позиційні та непозиційні системи числення | 2 | л |
| 3<br>4 | Позиційні і непозиційні системи числення. Трансляція чисел з однієї системи числення в іншу | 2 | пр |
| 5<br>6 | Методи опосередковування. Дії над двійковими числами | 2 | л |
| срс | Системи числення. Трансляція чисел з однієї системи числення в іншу | 2 | сам. роб |
| 7<br>8 | Операції над числами в двійковій системі числення | 2 | пр |
| срс | Операції над числами в двійковій системі числення | 2 | сам. роб |
| 9<br>10 | Логічні величини та логічні операції: кон'юнкція, диз'юнкція, імплікація, заперечення | 2 | л |
| 11<br>12 | Побудова таблиць істинності. Формули алгебри висловлень | 2 | пр |
| срс | Елементи алгебри логіки | 2 | сам. роб |
| 13<br>14 | Основи алгоритмізації. Властивості алгоритмів. Блок-схема | 2 | л |
| 15<br>16 | Основні способи запису алгоритмів | 2 | пр |
| 17<br>18 | Модульна контрольна робота | 2 | |
| **Змістовий модуль 2. Основи мови програмування Паскаль та методологія програмування** | | | |
| 19<br>20 | Основні етапи розв'язування задачі з використанням ЕОМ. Поняття програми. Мова програмування Паскаль. Поняття про інтерпретацію і компіляцію програм | 2 | л |
| 21<br>22 | Ознайомлення з середовищем програмування Turbo Pascal. Редагування тексту | 2 | л |
| 23<br>24 | Робота в середовищі програмування. Запуск програм на виконання | 2 | пр |

| *№ заняття* | *Тема заняття* | *К-ть год.* | *Тип заняття* |
|---|---|---|---|
| срс | Робота в середовищі програмування | 2 | сам. роб |
| 25<br>26 | Мова програмування Паскаль: алфавіт, синтаксис | 2 | л |
| 27<br>28 | Опис програм за правилами мови програмування | 2 | пр |
| срс | Типи даних і виразів в ТР, оператори мови Pascal | 2 | сам. роб |
| 29<br>30 | Поняття оператора: оператори введення-виведення, присвоєння | 2 | л |
| 31<br>32 | Створення і реалізація програм на введення і виведення даних | 2 | пр |
| срс | Оператори введення-виведення, присвоєння | 2 | сам. роб |
| 33<br>34 | Створення і реалізація лінійних програм | 2 | пр |
| 35<br>36 | Модульна контрольна робота | 2 | |
| ірс | Основи мови програмування Паскаль та методологія програмування | 6 | інд. роб |

## Лекція. Поняття про системи числення. Позиційні та непозиційні системи числення

**Мета, завдання лекції:** ознайомлення з поняттям системи числення, видами систем числення: позиційною та непозиційною; вивчення правил переведення чисел з однієї системи числення в іншу.

**План і організаційна структура лекції:**

1. Система числення: основні поняття.
2. Позиційна система числення.
3. Непозиційна система числення.
4. Правила переведення з десяткової системи числення в задану.
5. Правила переведення з заданої системи числення в десяткову.

**Зміст лекційного матеріалу:**

### *Системи числення. Основні поняття*

Десять пальців рук – це перший пристрій для рахунку, яким людина користується з доісторичних часів. Число 10 стало основою десяткової системи числення.

**Система числення** – це алфавіт системи та правила утворення чисел і дій з ними.

**Алфавіт системи числення** – це усі цифри, за допомогою яких утворюються числа певної системи числення. Алфавіт десяткової системи числення складається з десяти цифр: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

У різні історичні періоди користувалися різними системами числення, які були відмінними від десяткової. Наприклад, походження дванадцяткової системи пов'язують з рахунком на чотирьох пальцях, які мають 12 фаланг. До нас ця система дійшла у словах:

*дюжина* – 12
*грос* – дюжина дюжин – $12 \cdot 12 = 144 = 12^2$
*маса* – дюжина гросів – $12 \cdot 12 \cdot 12 = 1728 = 12^3$

Наприклад, це використовується в мірах довжини, грошових одиницях:
1 фут = 12 дюймів, 1 шилінг = 12 песо.

*Задача.* В містечку проживало 3 маси і 4 гроси людей. Яка це кількість людей в десятковій системі числення?

*Розв'язання:* 3 маси і 4 гроси = $3 \cdot 12^3 + 4 \cdot 12^2 = 5184 + 576 = 5760$.

*Відповідь:* В містечку проживало 5760 чоловік.

*Задача.* Два місяці. Скільки це приблизно декад?

*Розв'язання:* Позначимо кількість декад $x$.

1 місяць $\approx$ 30 днів, 1 декада = 10 днів. 2 місяці $\approx$ 60 днів. $x = 60/10 = 6$.

*Відповідь:* Два місяці – це приблизно 6 декад.

### *Позиційні та непозиційні системи числення*

Системи числення поділяються на позиційні і непозиційні.

**У позиційній системі числення** значення цифри залежить від позиції, яку цифра займає в зображенні числа. У цілих числах позиції нумеруються справа наліво

починаючи з нуля.

**5842**
3 2 1 0

Наприклад, у числі 5842 остання цифра 2 перебуває в нульовій позиції та означає кількість одиниць; передостання цифра 4 перебуває в першій позиції й означає кількість десятків і т.д.

Отже, число 5842 можна записати у вигляді такої суми:

$5\cdot1000 + 8\cdot100 + 4\cdot10 + 2\cdot1 = 5\cdot10^3 + 8\cdot10^2 + 4\cdot10^1 + 2\cdot10^0$.

Щоб означити, що число належить саме до десяткової системи числення, пишуть $(5842)_{10}$. Число 10 є **основою** десяткової системи числення.

Уявімо, що в людини не по 5, а по 4 пальці на кожній руці. Тоді б, очевидно, в нас виникла вісімкова система числення. Тобто, алфавіт такої системи буде складатися з цифр: 0, 1, 2, 3, 4, 5, 6, 7. Число 8 буде основою вісімкової системи числення.

Відповідність між вісімковою і десятковою системами числення така:

| вісімкова | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 20 | … | 100 | … | 144 | … | 1000 | … | 1750 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| десяткова | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | … | 64 | … | 100 | … | 512 | … | 1000 |

$(20)_8=(16)_{10}$; $(100)_8=(64)_{10}$; $(1000)_8=(512)_{10}$; $(144)_8=(100)_{10}$; $(1750)_8=(100)_{10}$.

Очевидно, що може існувати не лише вісімкова система числення, а й інші, наприклад двадцяткова, шістнадцяткова, двійкова. Наприклад, алфавіт шістнадцяткової системи: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Відповідність між шістнадцятковою і десятковою системами числення така:

| шістнадцяткова | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | A | B | C | D | E | F |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| десяткова | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |

### *Непозиційна система числення*

У непозиційній системі числення значення цифри не залежить від позиції, в якій вона розміщена у даному числі. Прикладом такої системи є римська.

***Алфавіт римської системи числення:***

**I** – один, **V** – п’ять, **X** – десять, **L** – п’ятдесят, **C** – сто, **D** – п’ятсот, **M** – тисяча.

Довільне число утворюють, комбінуючи цифри алфавіту. Щоб обчислити десяткове значення римського числа, потрібно додати значення всіх римських цифр. Цифри в римському числі розміщені, як звичайно, в порядку спадання їхніх значень.

Однак є виняток: якщо менша цифра стоїть перед більшою, тоді від більшого числа віднімають менше. Перед більшою цифрою може бути тільки одна менша цифра. Допустимі такі комбінації:

CM = 900, CD = 400, XD = 90, XL = 40, IX = 9, IV = 4.

*Приклади:*

XX = 10+10=20

LVIII = 50+5+3=58

XCIV = 90+4=94

XLIX = 40+9=49

*Примітка:* IL – недопустима комбінація.

### *Переведення чисел $(\ )_p \rightarrow (\ )_{10}$*

Відомо, що $(125)_{10}=(7D)_{16}$. Як же здійснювати перехід з певної системи числення в десяткову і навпаки?

Для цього необхідно абстрагувати систему числення, яка відмінна від десяткової, тобто подати її в узагальненому вигляді. Такою узагальненою системою є система числення з основою ***p***. Алфавіт такої системи: 0, 1, …, p-1.

Наприклад, основа п'ятіркової системи числення – **p=5**.

Алфавіт цієї системи: 0, 1, … 5-1, тобто: 0, 1, 2, 3, 4.

Перехід числа з системи з основою числення *p* в десяткову здійснюється за такою формулою:

$$(a\ a\ a\ \ldots\ aaa)_p = (a\cdot p^k + a\cdot p^{k-1} + a\cdot p^{k-2} + \ldots + a\cdot p^2 + a\cdot p^1 + a\cdot p^0)_{10}$$

Раніше ми записали, що $(144)_8=(100)_{10}$. Давайте перевіримо:

$$(144)_8 = 1\cdot 8^2 + 4\cdot 8^1 + 4\cdot 8^0 = 1\cdot 64 + 4\cdot 8 + 4\cdot 1 = 64+32+4 = (100)_{10}$$

2 1 0

Отже, для того, щоб перевести ціле число з системи, що має основу *p*, в десяткову систему числення, потрібно:

1. Проставити під цифрами даного числа справа наліво номер позиції починаючи з 0.
2. а) цифру найстаршого розряду числа з основою *p* (нехай це *a*) записати після знаку дорівнює;
   б) поставити знак множення;
   в) записати *p*;
   г) над числом *p* поставити степінь – номер позиції записаної цифри *a*;
   д) поставити знак плюс.
3. Виконати пункт 2 для цифри молодшого розряду.
4. Виконувати пункт 2, аж поки не дійдемо до цифри, позиція якої 0.
5. Виконати обчислення, записати число, поставити основу 10.

*Задача.* Перевести п'ятіркове число $(2413)_5$ в десяткове.

*Розв'язання.* $(2413)_5=2\cdot 5^3+4\cdot 5^2+1\cdot 5^1+3\cdot 5^0=250+100+8=(358)_{10}$

*Задача.* Перевести шістнадцяткове число $(7D)_{16}$ в десяткове.

*Розв'язання.* $(7D)_{16}=7\cdot 16^1+13\cdot 16^0=112+13=(125)_{10}$

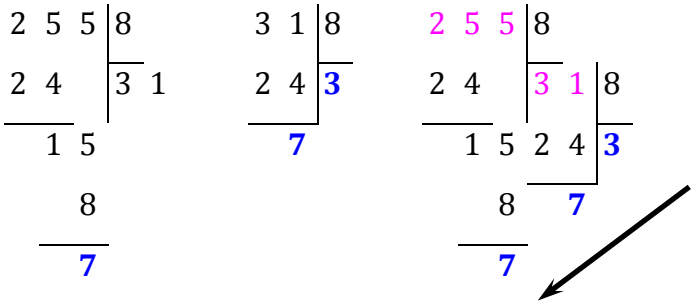
### *Переведення чисел $(\ )_{10} \rightarrow (\ )_p$*

Для того, щоб перевести ціле число з десяткової системи в систему з основою *p*, потрібно:

1. Десяткове число поділити на *p*.
2. Визначити остачу і частку.
3. Якщо частка менша від *p*, то виконати пункт 6, якщо частка більша від *p*, то виконати пункт 4.
4. Розглянути частку як нове число.
5. Виконати пункти 1, 2, 3.
6. Прочитати результат.

**Результат** – це послідовність цифр, що складається з останньої частки та всіх остач, починаючи від останньої.

Наприклад:

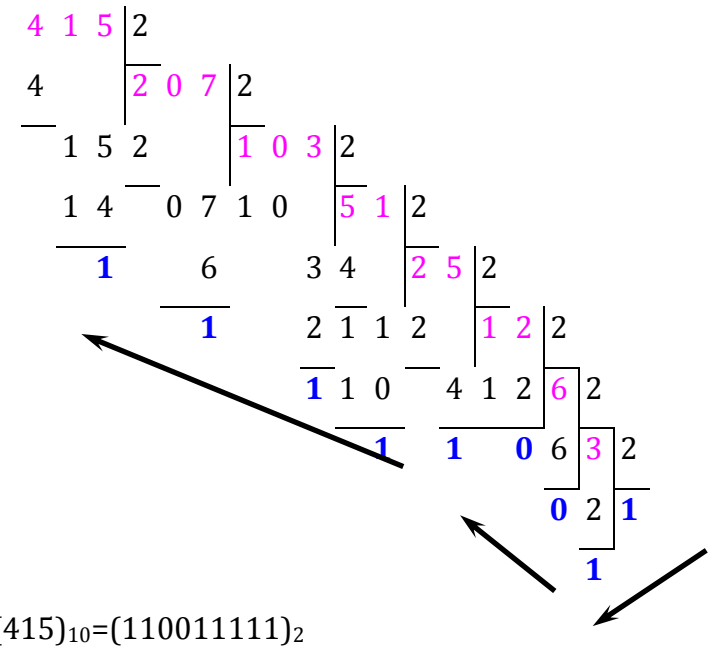


$(255)_{10}=(377)_8$

*Виконуємо перевірку:* $(377)_8=3\cdot8^2+7\cdot8^1+7\cdot8^0=192+56+7=(255)_{10}$.

*Задача.* Перевести десяткове число $(415)_{10}$ в двійкове.

*Розв'язання:*



$(415)_{10}=(110011111)_2$

*Виконуємо перевірку:*

$(100011111)_2=1\cdot2^8+1\cdot2^7+0\cdot2^6+0\cdot2^5+1\cdot2^4+1\cdot2^3+1\cdot2^2+1\cdot2^1+1\cdot2^0=256+128+0+0+16+8+4+2+1=(415)_{10}$.

## Практична робота. Позиційні і непозиційні системи числення. Трансляція чисел з однієї системи числення в іншу

**Мета:** формування навичок переведення чисел з однієї системи числення в іншу.

**Завдання:**

- *студенти повинні знати:* поняття системи числення; позиційні та непозиційні системи числення;
- *студенти повинні вміти:* виконувати переведення чисел з однієї системи числення в іншу: з десяткової в римську та навпаки, з заданої системи числення в десяткову та навпаки.

– **Обладнання:** дошка, зошит.

**Хід виконання:**

Здійснити переведення:

| **В-1** | **В-2** |
|---|---|
| **1.** $(363)_{10} \rightarrow (\ldots)_{рим}$;<br>$(824)_{10} \rightarrow (\ldots)_{рим}$;<br>$(513)_{10} \rightarrow (\ldots)_{рим}$.<br>**2.** $(1022)_{3} \rightarrow (\ldots)_{10}$<br>$(2A74,12)_{11} \rightarrow (\ldots)_{10}$<br>$(2AB7,15)_{15} \rightarrow (\ldots)_{10}$<br>$(245,56)_{8} \rightarrow (\ldots)_{10}$<br>$(110010,11)_{2} \rightarrow (\ldots)_{10}$<br>$(15,024)_{6} \rightarrow (\ldots)_{10}$<br>**3.** $(29)_{10} \rightarrow (\ldots)_{2}$<br>$(0,38)_{10} \rightarrow (\ldots)_{2}$<br>$(401)_{10} \rightarrow (\ldots)_{12}$<br>$(267,19)_{10} \rightarrow (\ldots)_{5}$ | **1.** $(456)_{10} \rightarrow (\ldots)_{рим}$;<br>$(2113)_{10} \rightarrow (\ldots)_{рим}$;<br>$(809)_{10} \rightarrow (\ldots)_{рим}$.<br>**2.** $(1201)_{3} \rightarrow (\ldots)_{10}$<br>$(37A3,12)_{11} \rightarrow (\ldots)_{10}$<br>$(B04,A1)_{15} \rightarrow (\ldots)_{10}$<br>$(413,24)_{8} \rightarrow (\ldots)_{10}$<br>$101101,11)_{2} \rightarrow (\ldots)_{10}$<br>$(23,055)_{6} \rightarrow (\ldots)_{10}$<br>**3.** $(31)10 \rightarrow (\ldots)_{2}$<br>$(0,76)_{10} \rightarrow (\ldots)_{2}$<br>$(503)_{10} \rightarrow (\ldots)_{12}$<br>$(354,82)_{10} \rightarrow (\ldots)_{5}$ |
| **В-3** | **В-4** |
| **1.** $(512)_{10} \rightarrow (\ldots)_{рим}$;<br>$(3015)_{10} \rightarrow (\ldots)_{рим}$;<br>$(916)_{10} \rightarrow (\ldots)_{рим}$.<br>**2.** $(1102)_{3} \rightarrow (\ldots)_{10}$<br>$42A15,26)_{11} \rightarrow (\ldots)_{10}$<br>$2B16,A4)_{15} \rightarrow (\ldots)_{10}$<br>$(627,23)_{8} \rightarrow (\ldots)_{10}$ | **1.** $(386)_{10} \rightarrow (\ldots)_{рим}$;<br>$(2803)_{10} \rightarrow (\ldots)_{рим}$;<br>$(472)_{10} \rightarrow (\ldots)_{рим}$.<br>**2.** $(2011)_{3} \rightarrow (\ldots)_{10}$<br>$23A14,62)_{11} \rightarrow (\ldots)_{10}$<br>$12A5,B7)_{15} \rightarrow (\ldots)_{10}$<br>$(574,73)_{8} \rightarrow (\ldots)_{10}$ |

| | |
|---|---|
| $101001,01)_2 \rightarrow (\ldots)_{10}$ | $110001,11)_2 \rightarrow (\ldots)_{10}$ |
| $(42, 035)_6 \rightarrow (\ldots)_{10}$ | $(35,024)_6 \rightarrow (\ldots)_{10}$ |
| **3.** $(42)_{10} \rightarrow (\ldots)_2$ | **3.** $(25)_{10} \rightarrow (\ldots)_2$ |
| $(0,89)_{10} \rightarrow (\ldots)_2$ | $(0,98)_{10} \rightarrow (\ldots)_2$ |
| $(324)_{10} \rightarrow (\ldots)_{12}$ | $(567)_{10} \rightarrow (\ldots)_{12}$ |
| $(267,93)_{10} \rightarrow (\ldots)_5$ | $(315,29)_{10} \rightarrow (\ldots)_5$ |

***Тренувальні вправи (переведення в римську сч та навпаки):***

| Десяткова сч | Римська сч | Десяткова сч | Римська сч |
|---|---|---|---|
| 345 | CCCXLV | 617 | DCXVII |
| 789 | DCCLXXXIX | 2458 | MMCDLVIII |
| 1987 | MCMLXXXVII | 145 | CXLV |
| 1238 | MCCXXXVIII | 784 | DCCLXXXIV |
| 1817 | MDCCCXVII | 1613 | MDCXIII |
| 3484 | MMMCDLXXXIV | 484 | CDLXXXIV |
| 784 | DCCLXXXIV | 569 | DLXIX |
| 345 | CCCXLV | 139 | CXXXIX |
| 173 | CLXXIII | 469 | CDLXIX |
| 2098 | MMXCVIII | 944 | CMXLIV |
| 374 | CCCLXXIV | 806 | DCCCVI |
| 667 | DCLXVII | 274 | CCLXXIV |
| 1775 | MDCCLXXV | | |

***Контрольні запитання:***

1. Система числення: основні поняття.
2. Позиційна система числення.
3. Непозиційна система числення.
4. Правила переведення з десяткової системи числення в задану.
5. Правила переведення з заданої системи числення в десяткову.

## Лекція. Методи опосередковування. Дії над двійковими числами

**Мета, завдання лекції:** ознайомлення з можливостями переведення чисел з однієї системи числення в іншу через опосередкування у вісімковій та шістнадцятковій системах числення; вивчення правил виконання обчислень в двійковій системі числення.

**План і організаційна структура лекції:**

1. Переведення з двійкової системи числення в десяткову через опосередкування в вісімковій системі числення.
2. Переведення з десяткової системи числення в двійкову через опосередкування в вісімковій системі числення.
3. Дії над двійковими числами.

**Зміст лекційного матеріалу:**

На практиці ми побачили, що перехід типу $(\ )_{10} \Rightarrow (\ )_2$ і навпаки $(\ )_{10} \Rightarrow (\ )_2$ вимагають громіздких обчислень.

Переведення, в яких число однієї системи числення з певною основою переводиться в число іншої системи числення з іншою основою, називається **прямим** або **безпосереднім**: $(\ )_n \Rightarrow (\ )_m$ і навпаки $(\ )_m \Rightarrow (\ )_n$.

**Непрямим** або ж **опосередкованим** переведенням називається таке переведення, коли: $(\ )_n \Rightarrow (\ )_s$, а потім $(\ )_s \Rightarrow (\ )_m$ і навпаки: $(\ )_m \Rightarrow (\ )_s$, а потім $(\ )_s \Rightarrow (\ )_n$.

Найчастіше опосередкування проводиться у двох системах:

1) опосередкування в вісімковій системі числення;

2) опосередкування в шістнадцятковій системі числення.

Для опосередкованого переведення нам потрібно знати **таблицю тріад** – таблиця відповідності вісімкових цифр і груп двійкових цифр:

У цій таблиці для вісімкових цифр 0, 1, 2, 3 в відповідних групах двійкових цифр спереду дописані нулі, яких відкидати не можна.

Для опосередкування переведення нам потрібно знати таблицю відповідності

| вісімкові цифри | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
|---|---|---|---|---|---|---|---|---|
| група двійкових цифр | 000 | 001 | 010 | 011 | 100 | 101 | 110 | 111 |

шістнадцяткових цифр і груп двійкових цифр:

| шістнадцяткові цифри | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** |
|---|---|---|---|---|---|---|---|---|
| група двійкових цифр | 0000 | 0001 | 0010 | 0011 | 0100 | 0101 | 0110 | 0111 |
| шістнадцяткові цифри | **8** | **9** | **A** | **B** | **C** | **D** | **E** | **F** |
| група двійкових цифр | 1000 | 1001 | 1010 | 1011 | 1100 | 1101 | 1110 | 1111 |

У цій таблиці для шістнадцяткових цифр 0, 1, 2, 3, 4, 5, 6, 7 у відповідних групах двійкових цифр спереду дописані нулі, яких відкидати не можна.

### *Переведення з двійкової системи числення в десяткову через опосередкування в вісімковій системі числення*

**І**. Переводимо число з двійкової системи числення в вісімкову:

а) розбити запис двійкового числа справа наліво по 3 цифри;

б) доповнити зліва нулями до 3 цифр крайній лівий запис;

в) кожну групу з 3 двійкових цифр замінити відповідною вісімковою цифрою.

г) записати результат $(\ )_2 \Rightarrow (\ )_8$.

**II.** Переводимо число з вісімкової системи числення в десяткову за правилом переходу

$(\ )_p \Rightarrow (\ )_{10}$.

*Задача.* Перевести $(1011001100111)_2$ в десяткову систему числення через опосередкування в вісімковій.

*Розв'язання:*

1) $001/011/001/100/111)_2 = (13147)_8$

2) $(13148)_8=1\cdot8^4+3\cdot8^3+1\cdot8^2+4\cdot8^1+7\cdot8^0 = 4096+1536+64+32+7=(5735)_{10}$

*Відповідь:* $(1011001100111)_2=(5735)_{10}$

*Задача.* Перевести $(11001001001000)_2$ в десяткову систему числення через опосередкування в вісімковій.

*Розв'язання:*

1) $(011/001/001/001/000)_2 = (31110)_8$

2) $(31110)_8=3\cdot8^4+1\cdot8^3+1\cdot8^2+1\cdot8^1+0\cdot8^0=12288+512+64+8+0=(12872)_{10}$

*Відповідь:* $(11001001001000)_2=(12872)_{10}$

***Переведення з десяткової системи числення в двійкову через опосередкування в вісімковій системі числення***

**I.** Переводимо число з десяткової системи числення в вісімкову за правилом переходу $(\ )_{10} \Rightarrow (\ )_p$.

**II.** Переводимо число з вісімкової системи числення в двійкову:

а) замінити кожну цифру вісімкового числа на відповідну групу двійкових цифр;

б) якщо зліва (тобто спереду) з'явилися нулі, то їх можна відкинути (нулі в кінці числа відкидати не можна);

в) записати результат $(\ )_8 \Rightarrow (\ )_2$.

*Задача.* Перевести $(3521)_{10}$ в двійкову систему числення через опесередкування в вісімковій.

*Розв'язання:*

1)

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | 3 | 5 | 2 | 1 | \| | 8 | | | | |
| | 3 | 2 | | | \| | 4 | 4 | 0 | \| | 8 |
| | | 3 | 2 | | | 4 | 0 | | \| | 5 | 5 | \| | 8 |
| | | 3 | 2 | | | | 4 | 0 | 4 | 8 | \| | **6** |
| | | | 0 | 1 | | | 4 | 0 | | **7** | | |
| | | | | **1** | | | | **0** | | | | |

$(3521)_{10}=(6701)_8$

2) $(6701)_8=(110/111/000/001)_2$

*Відповідь:* $(3521)_{10}=(110111000001)_2$

Алгоритм переведення чисел з двійкової системи числення в десяткову через опосередкування в шістнадцятковій системі числення і алгоритм переведення чисел з

десяткової системи числення в двійкову через опосередкування в шістнадцятковій системі числення аналогічні вище розглянутим алгоритмам.

***Завдання:***

**1**. Перевести $(111000100110011)_2$, $(100111100101100)_2$ в десяткову систему числення через опосередкування:

а) в вісімковій; б) в шістнадцятковій системі числення.

**2.** Перевести $(7328)_{10}$; $(9517)_{10}$ в двійкову систему числення через опосередкування

а) в вісімковій б) в шістнадцятковій системах числення.

### *Дії над двійковими числами*

При виконанні дії додавання і множення використовують таблиці додавання і множення двійкових чисел:

| |
|---|
| 0 + 0 = 0 |
| 0 + 1 = 1 |
| 1 + 0 = 1 |
| 1 + 1 = 10 |

| |
|---|
| 0 · 0 = 0 |
| 0 · 1 = 0 |
| 1 · 0 = 0 |
| 1 · 1 = 1 |

Пам'ятайте, що$(1)_2=(1)_{10}$, але $(10)_2=(2)_{10}$.

Тобто допис 1+1=10 означає, що в нульовому розряді результату пишемо цифру нуль, а одиницю переносимо у старший розряд.

*Приклади:*

1)
$$\begin{array}{r} 1\ 1\ 0\ 1\ 0\ 1 \\ +\quad 1\ 0\ 1\ 1 \\ \hline 1\ 0\ 0\ 0\ 0\ 0\ 0 \end{array}$$

2)
$$\begin{array}{r} 1\ 1\ 0\ 0\ 1\ 1 \\ +\ 1\ 0\ 1\ 1\ 1\ 0 \\ \hline 1\ 1\ 0\ 0\ 0\ 0\ 1 \end{array}$$

3)
$$\begin{array}{r} 1\ 1\ 1\ 1 \\ +\ 1\ 1\ 1\ 1 \\ \hline 1\ 1\ 1\ 1\ 0 \end{array}$$

4)
$$\begin{array}{r} 1\ 0\ 1\ 1 \\ *\quad 1\ 0\ 1 \\ \hline 1\ 0\ 1\ 1 \\ 0\ 0\ 0\ 0 \\ 1\ 0\ 1\ 1 \\ \hline 1\ 1\ 0\ 1\ 1\ 1 \end{array}$$

5)
$$\begin{array}{r} 1\ 1\ 0\ 1\ 1 \\ *\quad 1\ 0\ 0 \\ \hline 0\ 0\ 0\ 0\ 0 \\ 0\ 0\ 0\ 0\ 0 \\ 1\ 1\ 0\ 1\ 1 \\ \hline 1\ 1\ 0\ 1\ 1\ 0\ 0 \end{array}$$

6)
$$\begin{array}{r} 1\ 1\ 1\ 1 \\ \quad 1\ 1\ 1 \\ \hline 1\ 1\ 1\ 1 \\ 1\ 1\ 1\ 1 \\ 1\ 1\ 1\ 1 \\ \hline 1\ 1\ 0\ 1\ 0\ 0\ 1 \end{array}$$

Слід бути дуже уважними при переході через розряд.

Потрібно пам'ятати наступне:

1+1=10

1+1+1=10+1=11

1+1+1+1=10+10=100

1+1+1+1+1=10+10+1=101

1+1+1+1+1+1=10+10+10=110

1+1+1+1+1+1+1=10+10+10+1=111 і т.д.

Аналізуючи приклади множення двох чисел у двійковій системі числення, можна

стверджувати, що операція множення зводиться до поетапного додавання множеного.

При виконанні дії віднімання використовують таблицю віднімання двійкових чисел:

| |
|---|
| 0 – 0 = 0 |
| 1 – 0 = 1 |
| 1 – 1 = 0 |
| 10 – 1 = 1 |

*Приклади:*

1)
$$\begin{array}{r} 1\,0\,0\,1\,1\,1\,0\,1 \\ -\quad\quad\quad 1\,0\,1\,1 \\ \hline 1\,0\,0\,1\,0\,0\,1 \end{array}$$

2)
$$\begin{array}{r} 1\,0\,0\,0 \\ -\quad\quad 1\,1\,1 \\ \hline 0\,0\,0\,1 \end{array}$$

3)
$$\begin{array}{r} 1\,0\,1\,0\,1 \\ -\quad\quad 1\,1\,1\,1 \\ \hline 0\,1\,1\,0 \end{array}$$

### *Ділення двійкових чисел*

Ділення двійкових чисел в стовпчик виконується аналогічно, як і в десятковій системі числення.

*Приклади:*

1)
$$\begin{array}{r|l} 1\,1\,1\,1\,0 & 1\,0\,1 \\ \cline{2-2} -\,1\,0\,1 & 1\,1\,0 \\ \cline{1-1} \quad 1\,0\,1 & \\ -\,1\,0\,1 & \\ \cline{1-1} \quad\quad 0 & \end{array}$$

2)
$$\begin{array}{r|l} 1\,0\,0\,1\,1\,1 & 1\,1\,0 \\ \cline{2-2} -\,1\,1\,0 & 1\,1\,0,\,1 \\ \cline{1-1} \quad 1\,1\,1 & \\ -\,1\,1\,0 & \\ \cline{1-1} \quad\quad 1\,1\,0 & \\ -\quad 1\,1\,0 & \\ \cline{1-1} \quad\quad\quad 0 & \end{array}$$

*Виконуємо перевірку:*

1)
$$\begin{array}{r} 1\,0\,1 \\ *\quad 1\,1\,0 \\ \hline 0\,0\,0 \\ 1\,0\,1 \\ 1\,0\,1 \\ \hline 1\,1\,1\,1\,0 \end{array}$$

2)
$$\begin{array}{r} 1\,1\,0,\,1 \\ *\quad\quad 1\,1\,0 \\ \hline 0\,0\,0\,0 \\ 1\,1\,0\,1 \\ 1\,1\,0\,1 \\ \hline 1\,0\,0\,1\,1\,1,\,0 \end{array}$$

## Практична робота. Операції над числами в двійковій системі числення

**Мета:** формування навичок виконання дій з двійковими числами.

**Завдання:**

- *студенти повинні знати:* поняття системи числення; позиційні та непозиційні системи числення; правила виконання обчислень в двійковій системі числення;
- *студенти повинні вміти:* виконувати арифметичні дії над двійковими числами.
- **Обладнання:** дошка, зошит.

**Хід виконання:**

Виконати арифметичні дії в двійковій системі числення:

**В-1**

**а)** 100101 + 111101

**б)** 1100010111+10011

**в)** 10011*110

**г)** 110101 – 101101

**д)** 100011 : 101

**В-2**

**а)** 1011011 + 1111011

**б)** 110010100 +1110

**в)** 10011*111

**г)** 1101011 – 10111

**д)** 100111 : 111

**В-3**

**а)** 1001011 + 1111011

**б)** 1001110111 +1111

**в)** 10011*101

**г)** 1101011 – 1011010

**д)** 101010 : 110

***Контрольні запитання:***

**1.** Таблиці додавання та віднімання двійкових чисел.
**2.** Множення двійкових чисел.
**3.** Ділення двійкових чисел

## Лекція. Логічні величини та логічні операції: кон'юнкція, диз'юнкція, імплікація, заперечення

**Мета, завдання лекції:** ознайомити студентів з поняттям математичної логіки, алгебри висловлень, вивчити основні логічні операції і дії над ними, формули алгебри висловлень, основні правила побудови таблиць істинності, ввести поняття еквівалентних висловлювань, вивчити основні властивості логічних операцій.

**План і організаційна структура лекції:**

1. Основні поняття математичної логіки.
2. Логічні операції.
3. Формули алгебри висловлень.
4. Таблиці істинності.

**Зміст лекційного матеріалу:**

**Математична логіка** – наука про закони людського мислення.

**Математична логіка** – це розділ математики, в якому закони мислення, зокрема в математиці, досліджуються математичними методами.

Ідея побудови логіки на основі математичних законів належить німецькому вченому Лейбніцу. Він вважав, що основні поняття логіки треба позначати символами, які з'єднуються між собою за певними правилами, і це дасть можливість висловлювання замінити обчисленнями.

Наприклад, запишемо твердження**:** **«Всі люди – смертні»:** Всі **S є P**

**«Сократ – людина»: X є S**

Висновок: **«Сократ – смертний»: X є P**

Ідею Лейбніца продовжив німецький математик Джордж Буль (1815-1864), який побудував в 1847 році алгебру висловлень, в якій використовувались математичні методи.

Поряд з терміном **"алгебра логіки"** вживають терміни "**алгебра Буля"** та **"алгебра висловлень".**

Отже, **алгебру логіки** утворюють висловлення та логічні операції над ними.

**Висловленням** називається розповідне речення, яке стверджує певну закінчену думку; з приводу твердження можемо сказати, істинне воно чи хибне. Якщо висловлення відповідає дійсності, то воно є ***істинним***. Якщо ж висловлення не відповідає дійсності, то воно є ***хибним***.

Наприклад, висловлення «4 – парне число» – істинне;

«2 – просте число» – істинне

«27 – просте число» – хибне.

Висловлення повинно задовольняти 2 умовам::

a) воно не може бути істинним і хибним одночасно – *закон протиріччя*;

*b)* висловлення є обов'язково істинним або хибним – *закон виключення третього.*

Висловлення називається **простим** (елементарним), якщо будь-яка його частина не є висловленням. Складні висловлення будують з простих за допомогою простих зв'язок «і», «або», «якщо…то», «тоді і лише тоді», «ні» і ін.

Прості висловлення будемо позначати великими латинськими буквами

A,B, C,…, $A_1$, $A_2$,..

Є лише дві ***логічні сталі***: 0 (хибність) і 1 (істинність).

Якщо висловлення, позначене літерою А, є істинним, то пишуть А=1. Якщо висловлення, позначене літерою В, є хибним, то пишуть В=0.

***Логічні операції*** (логічні зв'язки, з допомогою яких з простих висловлень отримуються складні): кон'юнкція, диз'юнкція, імплікація, еквівалентність, заперечення.

**Кон'юнкцією** висловлень. (&, Λ, «і») Р та Q називається складне висловлення, яке позначається PΛQ (читається «Р та Q» або «кон'юнкція Р та Q»), яке істинне тоді і тільки тоді, коли Р та Q істинні і хибне у всіх інших випадках.

Залежність істинності можна описати у вигляді таблиці істинності:

| **P** | **Q** | **PΛQ** |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**Диз'юнкцією** висловлень (V, +, «або») Р та Q називається складне висловлення, яке позначається PVQ (читається «Р або Q» або «диз'юнкція Р та Q»), яке хибне тоді і тільки тоді, коли Р та Q хибні одночасно.

| **P** | **Q** | **PVQ** |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

**Імплікацією** висловлень (→, «якщо…то») Р та Q називається складне висловлення, яке позначається P→Q (читається «Р імплікує Q»), яке хибне тоді, коли Р – істинне і Q – хибне і істинне в усіх інших випадках.

| **P** | **Q** | **P→Q** |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

Наприклад: «якщо число N націло ділиться на 6, то число N націло ділиться на 3»

**Еквівалентністю** висловлень (↔, «тоді і лише тоді», «необхідно і достатньо») Р та Q називається складне висловлення, яке позначається P↔Q (читається «Р еквівалентно Q»), яке істинне тоді і лише тоді, коли Р і Q – істинні або хибні одночасно.

| **P** | **Q** | **P↔Q** |
|---|---|---|
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

**Запереченням** висловлень ( ¯ , «ні») Р називається складне висловлення, яке позначається $\overline{P}$ (читається «не Р»), яке істинне, коли Р хибне і хибне, коли Р - істинне.

| P | $\overline{P}$ |
|---|---|
| 0 | 1 |
| 1 | 0 |

Приклад. *Мама сказала доньці: "Якщо ти зробила всі уроки і прибрала в квартирі, то можеш піти погуляти".*

Цей складний вислів складається з трьох простих. Два перших простих вказують на умову, а третій – на наслідок. Позначимо вислови (логічні змінні) великими латинськими літерами:

1) зробила уроки – A;
2) прибрала в квартирі – B;
3) можна іти гуляти – C.

Складають **таблицю істинності** логічної операції ***I*** (кон'юнкція).

| **X** | **Y** | **X∧Y** |
|---|---|---|
| 0 | 0 | ***0*** |
| 1 | 0 | ***0*** |
| 0 | 1 | ***0*** |
| 1 | 1 | ***1*** |

Приклад. *Мама сказала доньці: "Для того, щоб піти гуляти, тобі спочатку треба підмести подвір'я або помити посуд".*

Цей складний вислів складається з трьох простих. Перший простий вислів вказує на наслідок, а другий і третій на умови, одну з яких треба виконати. Позначимо вислови (логічні змінні) великими латинськими літерами:

1) можна іти гуляти – C;
2) підмела подвір'я – A;
3) помила посуд – B.

Складають **таблицю істинності** логічної операції ***АБО*** (диз'юнкція).

| **X** | **Y** | **X∨Y** |
|---|---|---|
| 0 | 0 | ***0*** |
| 1 | 0 | ***1*** |
| 0 | 1 | ***1*** |
| 1 | 1 | ***1*** |

Істинність складних висловлень залежить від істинності простих і від структури висловлення.

Наприклад, істинність висловлень $P \vee (Q \to R)$ та $(P \vee Q) \to R$ залежить від набору пропозиційних змінних та від послідовності виконання операцій.

***Завдання:*** Побудувати таблицю істинності:

a) $(\overline{P} \to Q) \vee (\overline{P \to Q})$
b) $P \vee (\overline{P} \wedge Q)$

**Формула алгебри висловлень** – це послідовність символів, побудованих за певними правилами, яка задає структуру складного висловлення.

Наприклад: $P \to (Q \wedge \overline{P})$ є формулою, а $P \to Q \wedge \leftrightarrow R$ не є формулою.

За аналогією в алгебрі:

(а + в) * с – правильно;

а + - в - неправильно.

Для спрощення формул алгебри висловлень приймемо такі узгодження:

1) зовнішні дужки, в які береться формула, можна не писати;
2) знак заперечення можна писати над формулою, до якої відноситься заперечення; і саму формулу в дужки можна не брати; $\overline{(\mathrm{P} \vee Q)}$ $\overline{P \vee Q}$
3) для зменшення кількості дужок приймемо такий пріоритет операцій: $\overline{\ }$, $\wedge$, $\vee$, $\rightarrow$, $\leftrightarrow$.

Завдання: побудувати таблицю істинності: $(\overline{\mathrm{P}} \rightarrow \mathrm{Q}) \vee (\overline{P \wedge R})$

**Типи формул**:

1. Формула, яка на всіх наборах значень пропозиційних змінних, які в неї входять, приймає значення істинно, наз. **тавтологією** або тотожно істинною формулою: $\models \alpha$.
2. Формула, яка на всіх наборах значень пропозиційних змінних приймає значення хибно, наз. **суперечністю**, або **тотожно хибною формулою**.

Ф-ли $\alpha, \beta$ наз. **рівносильними**, якщо на всіх наборах пропозиційних змінних, які входять в ці формули, ці формули приймають однакові значення. Записується:

$$\alpha \equiv \beta$$

## Практична робота. Побудова таблиць істинності. Формули алгебри висловлень

**Мета:** формувати у студентів навички побудов таблиць істинності; вміння використовувати властивості формул алгебри висловлень.

**Завдання:**

- *студенти повинні знати:* логічні операції; формули алгебри висловлень;
- *студенти повинні вміти:* виконувати побудову таблиць істинності логічних виразів.

**Обладнання:** дошка, зошит.

**Хід виконання:**

**В-1**

1. Побудувати таблицю істинності для формули $(\overline{P} \to Q) \vee (P \wedge Q)$.
2. Використовуючи побудову таблиць істинності, довести закон де Моргана: $\overline{P \wedge Q} \equiv \overline{P} \vee \overline{Q}$
3. Доведіть еквівалентність висловлювання, спростивши вираз:

   $\overline{A \vee \bar{A}} \vee \bar{U} \vee \overline{\bar{U}} = U$. Перевірте це, побудувавши таблиці істинності.

**В-2**

1. Побудувати таблицю істинності для формули $(\overline{P} \leftrightarrow Q) \wedge P \vee Q$
2. Використовуючи побудову таблиць істинності, довести закон де Моргана: $\overline{P \vee Q} \equiv \overline{P} \wedge \overline{Q}$
3. Доведіть еквівалентність висловлювання, спростивши вираз:

   $\overline{A \vee \bar{A}} \vee \overline{\bar{U} \vee \bar{U}} = U$. Перевірте це, побудувавши таблиці істинності.

***Контрольні запитання:***

1. Основні поняття математичної логіки.
2. Логічні операції, їх види.
3. Формули алгебри висловлень.
4. Таблиці істинності логічних операцій.
5. Види логічних елементів.

## Лекція. Основи алгоритмізації. Властивості алгоритмів. Блок-схема

**Мета, завдання лекції:** сформувати поняття про алгоритм, його властивості та способи подачі; навчити розпізнавати алгоритми навколо себе; формувати вміння розрізняти правильно та неправильно сформульовані алгоритми; дати поняття про базові структури алгоритмів; навчити розпізнавати базові структури в запропонованих алгоритмах, будувати основні типи блок-схем, вчити використовувати вказівку розгалуження і циклу в стандартних умовах.

**План і організаційна структура лекції:**

1. Поняття алгоритму.
2. Властивості алгоритмів.
3. Способи запису алгоритмів.
4. Базові структури алгоритмів.

**Зміст лекційного матеріалу:**

*Алгоритм* – чітко визначена послідовність дій, які спрямовані на досягнення поставленої мети або розв'язання задач певного типу.

*Приклад :* Обчислити $\frac{360}{92-32}-5$

Алгоритм: щоб розв'язати дану задачу, слід виконати таку послідовність дій:

1. Виконати віднімання 92-32 і запам'ятати результат (60).
2. Виконати ділення 360 : 60 і запам'ятати результат (6).
3. Виконати віднімання 6 – 5 і записати результат (1).

Кожна людина вже з семи років точно користується алгоритмами, не знаючи про це. План дня – це і є алгоритм.

***Базові структури (конструкції, види) алгоритмів:***

**1.** *Слідування (лінійний алгоритм)* **–** команди виконуються послідовно одна за одною.

**2.** *Розгалуження або вибір (є умова "Якщо")* **–** вказівка виконати одну із двох команд в залежності від того, чи виконується умова чи ні.

*Наприклад,* алгоритм переходу через дорогу:

1. Подивитися наліво.
2. Якщо немає перешкоди, то виконати пункт 8, інакше - пункт 3.
3. Пропустити автомобілі
4. Перейти дорогу до середини
5. Подивитися направо
6. Якщо немає перешкоди то виконати пункт 8 інакше пункт 7
7. Пропустити автомобілі
8. Завершити перехід через дорогу.

***3.*** *Повторення або цикл* – багаторазове повторення одних і тих самих дій (алгоритм ходьби, їзди на велосипеді).

***4.*** *Змішані алгоритми.*

Кожен алгоритм можна подати у вигляді комбінації трьох базових алгоритмічних структур.

***Властивості алгоритмів:***

**1.** *Зрозумілість* **–** в алгоритмі використовуються лише допустимі команди, ті, які розуміє виконавець.

**2.** *Скінченність* – алгоритм повинен завершуватись за певну кількість операцій**.**

**3.** *Результативність* – алгоритм має призвести до певного результату (навіть і до неправильного).

**4.** *Формальність* – виконавець повинен отримати результат, не вникаючи в його суть.

**5.** *Визначеність* – будь-який алгоритм повинен бути описаний так, щоб при його розшифруванні чи виконанні не виникло двозначності чи невизначеності.

**6.** *Масовість* – алгоритм повинен розв'язувати цілий клас задач (приклад – алгоритм переходу вулиці).

***Способи запису алгоритмів:***

***1.*** *Словесний.*

***2.*** *За допомогою навчальної алгоритмічної мови (НАМ).*

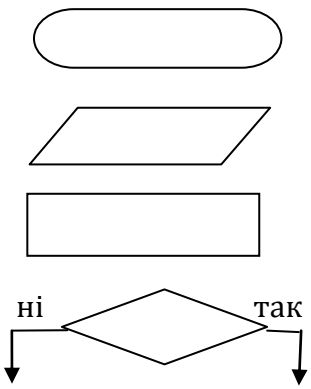
Ці мови мають чітко визначений синтаксис і максимально наближені до машинної мови (мови програмування). Але створені вони з навчальною метою, тому мають зрозумілий для людей вигляд. Таких псевдокодів зараз існує велика кількість, починаючи з графічних середовищ «Алгоритміка», «Роботландія», «Черепашка» тощо і закінчуючи текстовими «національними» реалізаціями алгоритмічних мов, подібних Паскалю. Ці псевдокоди мають програмну реалізацію і дуже широко застосовуються на етапі навчання основам програмування.

***3.*** *За допомогою певної мови програмування (у вигляді програми)*

На практиці найчастіше виконавцем створеного людиною алгоритму є комп'ютер і тому алгоритм має бути написаний мовою, зрозумілою для комп'ютера, тобто мовою програмування.

***4.*** *Графічний* (з допомогою блок-схеми).

Графічний спосіб подання алгоритмів був запропонований в інформатиці для наочності представлення алгоритму за допомогою набору спеціальних блоків. Основні з цих блоків такі:




– початок (кінець) алгоритму

– блок введення-виведення даних

– командний блок

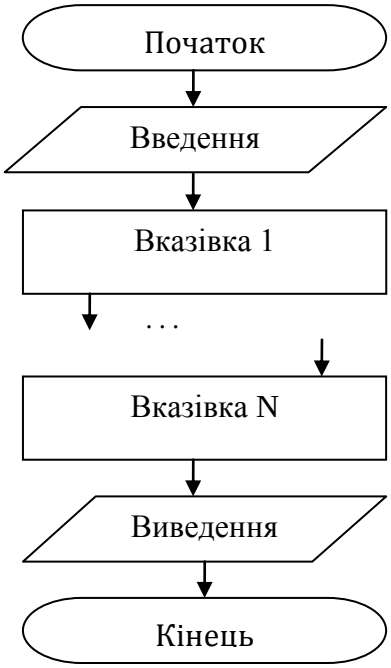
– блок умовного переходу (розгалуження)

*Виконавець алгоритму* – жива істота (людина чи тварина) або автоматичний пристрій (робот, верстат з програмним керуванням, електронна обчислювальна машина тощо), що спроможна діяти і діє згідно з наданим алгоритмом.

*Система вказівок виконавця* – сукупність усіх вказівок, які може виконувати даний виконавець.

Найпростіша в написанні та виконанні перша з структур алгоритмів – *лінійна*. До неї відносяться алгоритми, що складаються лише з простих команд. Простими є ті команди, що виконуються безумовно, тобто після першої команди виконується друга, потім третя і тощо.

Загальний вигляд лінійного алгоритму, поданий мовою блок-схем, наступний:



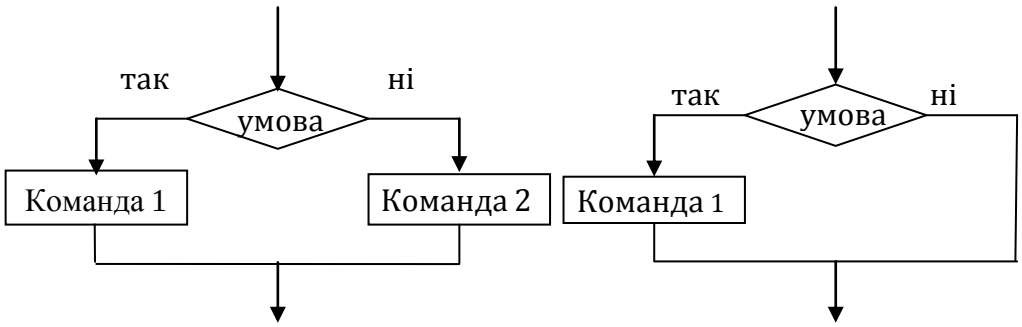

Лінійним можна назвати алгоритм ранкового збирання до школи:

- прокинутися;
- зробити ранковий туалет;
- одягнутися;
- поснідати;
- зібрати речі;
- одягнути верхній одяг,
- взутися;
- вийти до школи.

Набагато частіше зустрічається другий тип алгоритму – *розгалужений*. Цей алгоритм обов'язково містить в собі хоча б одну умову (як правило, їх набагато більше), і виконується він в залежності від цієї умови.

*Умовою* називається таке речення, на яке можна дати відповідь «так» чи «ні». Як правило, кажуть, що в першому випадку (коли ми відповіли на речення «так») умова є істинною, а в другому – хибною. Виходячи з цього, речення «Якого кольору твій піджак?» не можна вважати умовою, а речення «Твоє волосся русяве?» – можна.

Мовою блок-схем розгалужений алгоритм подається наступним чином:




**Повне розгалуження** **Неповне розгалуження**

*Повна форма* вказівки розгалуження виконується так:

- Якщо умова істинна, то виконується вказівка 1, а потім вказівка, яка знаходиться в програмі після всієї вказівки розгалуження;
- Якщо умова хибна, то виконується вказівка 2, а потім вказівка, яка знаходиться в програмі після всієї вказівки розгалуження.

*Скорочена форма* вказівки розгалуження виконується так:

- Якщо умова істинна, то виконується вказівка 1, а потім вказівка, яка знаходиться в програмі після всієї вказівки розгалуження;
- Якщо умова хибна, то виконується вказівка, яка знаходиться в програмі після вказівки розгалуження.

Однак, навіть маючи в своєму арсеналі команду розгалуження, важко реалізувати алгоритми, що потребують багаторазового повторення деякої послідовності однакових дій. Такі алгоритми називають *циклічними*.

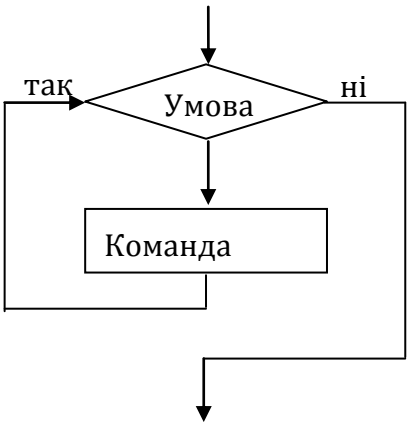
У нашому житті дуже часто зустрічаються алгоритми з повтореннями, причому чітко визначаються два типи повторів. В одному випадку ми чітко знаємо, скільки разів необхідно повторити задану послідовність команд, а в іншому – ні.

В залежності від того, чи знаємо ми, скільки разів необхідно повторювати якусь послідовність команд, розрізняють *цикли з лічильником* (кількість повторень відома заздалегідь) та *цикли з умовою* (цикл повторюється доти, доки не виконається якась умова).

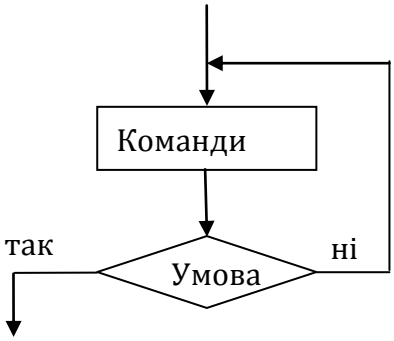
Крім того, в *циклах з умовою* теж можна виділити два різних випадки:

- *цикл з передумовою* – спочатку ми перевіряємо умову, а потім виконуємо деяку послідовність дій (так, ми спочатку перевіряємо, чи вимита підлога в тій класній кімнаті, де нам потрібно чергувати, а потім починаємо прибирання);
- *цикл з післяумовою* – спочатку ми виконуємо хоч один раз необхідну послідовність дій, а потім перевіряємо, чи не досягли ми бажаного результату (коли ми хочемо пити, ми спочатку робимо хоча б ковток води, а потім починаємо контролювати, чи не вгамували ми свою спрагу).

Мовою блок-схем обидва типи циклів виглядають наступним чином:




**Цикл з передумовою**




**Цикл з післяумовою**

Найчастіше ж у житті ми використовуємо *змішані алгоритми*, в яких поєднуються елементи лінійних, розгалужених та циклічних алгоритмів.

## Практична робота. Основні способи запису алгоритмів

**Мета:** формувати у студентів навички побудови основних типів блок-схем, використання вказівки розгалуження і циклу при створення словесного опису алгоритму розв'язування задачі та побудови блок-схеми.

**Завдання:**

- *студенти повинні знати:* поняття алгоритму, властивості алгоритмів, основні способи запису алгоритмів;
- *студенти повинні вміти:* використовувати словесний та графічний спосіб запису алгоритмів розв'язування задачі.

**Обладнання:** дошка, зошит.

**Хід виконання:**

**В-1**

1. Записати базові структури алгоритмів. Зобразити блок-схему повного та неповного розгалуження.
2. Скласти алгоритм розв'язування задачі. Записати його в словесній і графічній формі (за допомогою блок-схеми)
   ***Обчислити довжину гіпотенузи та площу прямокутного трикутника за заданими двома катетами.***
3. Ввести будь-яке значення x і обчислити значення складеної функції y, яка задана формулою:

$$y = \begin{cases} x^2 - 8, \ якщо \ x \geq 0, \\ 5 - \dfrac{4}{x}, \ якщо \ x < 0. \end{cases}$$

**В-2**

1. Записати властивості алгоритмів. Зобразити основні блоки, які використовують для побудови блок-схеми алгоритму.
2. Скласти алгоритм розв'язування задачі. Записати його в словесній і графічній формі (за допомогою блок-схеми)
   ***Обчислити катет та площу прямокутного трикутника за заданими гіпотенузою та другим катетом.***
3. Ввести будь-яке значення x і обчислити значення складеної функції y, яка задана формулою:

$$y = \begin{cases} 3x^2 - 8, \ якщо \ \ x \geq 5, \\ 4x - 3, \ якщо \ \ x < 5. \end{cases}$$

**В-3**

1. Записати базові структури алгоритмів. Побудувати блок-схеми для циклу з передумовою та післяумовою.
2. Скласти алгоритм розв'язування задачі. Записати його в словесній і графічній формі (за допомогою блок-схеми)
   ***Обчислити периметр і площу трикутника за трьома сторонами.***

3. Ввести будь-яке значення x і обчислити значення складеної функції y, яка задана формулою:

$$y = \begin{cases} \frac{x}{5} - 8, \ якщо \ x > 3, \\ 3x^2 - 4, \ якщо \ x \le 3. \end{cases}$$

***Додаткові завдання:***

**1.** Скласти алгоритм розв'язування задачі. Записати його в словесній і графічній формі (за допомогою блок-схеми)

***Знайти площу трапеції за відомими висотою і основами.***

**2.** Скласти алгоритм розв'язування задачі. Записати його в словесній і графічній формі (за допомогою блок-схеми)

***Швидкість світла 299792 км/с. Яку відстань долає світло за годину, добу?***

**3.** Скласти алгоритм розв'язування задачі. Записати його в словесній і графічній формі (за допомогою блок-схеми)

***Ділянка лісу має форму рівнобічної трапеції. Обчислити її периметр і площу за заданими сторонами.***

**4.** Скласти алгоритм розв'язування задачі. Записати його в словесній і графічній формі (за допомогою блок-схеми)

***Ресторан закуповує щодня масло $m_1$ кг по 50.50 грн. за кілограм, сметану $m_2$ кг по 8.40 грн., вершки $m_3$ кг по 12.10 грн. Визначити суми, потрібні для купівлі окремих продуктів, і загальну суму.***

**5.** Скласти алгоритм розв'язування задачі. Записати його в словесній і графічній формі (за допомогою блок-схеми)

***Телефонні розмови з трьома населеними пунктами коштують $c_1$, $c_2$, $c_3$ коп/хв. Розмови тривали $t_1$, $t_2$, $t_3$ хв відповідно. Яку суму нарахує комп'ютер до оплати за кожну і всі розмови?***

**6.** Скласти алгоритм розв'язування задачі. Записати його в словесній і графічній формі (за допомогою блок-схеми)

***Обчислити сторону та площу $S=d^2/2$ квадрата, якщо відома його діагональ d.***

## Лекція. Основні етапи розв’язування задачі з використанням ЕОМ. Поняття програми. Мова програмування Паскаль. Поняття про інтерпретацію і компіляцію програм

**Мета, завдання лекції:** ознайомити студентів з загальною схемою поетапного розв’язування задач з використанням готового програмного забезпечення (ПЗ) і за допомогою програмування; ввести поняття моделі, математичної моделі; ознайомити з поняттям програми, мови програмування, системи програмування, компіляції, інтерпретації.

**План і організаційна структура лекції:**

1. Основні етапи розв’язування задачі з використанням ЕОМ.
2. Поняття мови програмування, програми.
3. Класифікація мов програмування.
4. Система програмування.

**Зміст лекційного матеріалу:**

**Загальна схема процесу поетапного розв'язування задач з використанням ЕОМ:**

***Перший етап*** — постановка задачі.

Опис вхідних даних та умов, формулювання мети завдання, опис очікуваних результатів. При цьому встановлюються обмеження на можливі значення вихідних даних і необхідних результатів.

Навіщо накладати обмеження на результати? Наприклад, ви одержали у відповіді від'ємний час. Отже, під час розв'язання допущено помилку.

***Другий етап*** — побудова інформаційної моделі.

**Модель** — це штучно створений людиною абстрактний або матеріальний образ реального об’єкта, що зберігає типові властивості оригіналу, важливі для розв’язання певної задачі. У моделі використовуються тільки ті властивості і закономірності поведінки об'єктів, що є істотними для розв'язування задачі.

Математичні моделі описують об’єкти, процеси і явища за допомогою математичних співвідношень, формул, рівнянь (формули фізичних законів, рівняння хімічних реакцій).

Для побудови моделі слід:

- зрозуміти, в якій предметній області шукати описи об'єктів, що існують в умові задачі;
- відібрати описи, істотні для розв'язуваної задачі;
- із відібраних описів побудувати єдиний опис, що поєднує необхідні в задачі результати з вихідними даними, тобто опис, який забезпечує розв'язування поставленої задачі.

***Третій етап*** — розробка алгоритму та його комп’ютерна реалізація. Розробка алгоритму згідно з вибраним методом розв’язання задачі; визначення необхідного прикладного програмного забезпечення або розробка нового програмного забезпечення (вибір мови програмування, написання, налагодження та тестування програми); виконання програми із заданими вхідними даними.

***Четвертий етап*** — аналіз результатів. Аналіз результатів, що отримані на моделях та реальних об’єктах, для виправлення помилок і доопрацювання розробленого алгоритму та програми, що пройшла тестування на моделі.

Робота з розв’язання прикладної задачі з використанням комп’ютера проходить у кілька етапів, кожен із яких важливий по-своєму, і тільки правильне виконання кожного з етапів призводить до розв’язання задачі в цілому. Перед виконанням будь-якого проекту доцільно змоделювати хід виконання і спрогнозувати результати роботи. У цьому може допомогти побудова інформаційної моделі об’єкта-оригіналу або явища (процесу), що досліджується. В більшості випадків це дозволяє швидко і з невеликими витратами розрахувати результати виконання масштабної реальної задачі.

**Програма** – впорядкована послідовність дій для ЕОМ, виконання якої реалізує алгоритм розв'язку будь-якої задачі.

*Основні складові програми* – логіка, структури даних та інтерфейс. Логіка визначає поведінку програми, структури даних – спосіб зберігання даних, а інтерфейс є засобом взаємодії програми з користувачем та іншими програмами.

**Мова програмування** – це система позначень, за допомогою яких можна точно описати алгоритм для виконання його комп’ютером. Алгоритм, записаний мовою програмування, називається **програмою.** Мова програмування – це набір символів (*алфавіт*), система правил складання базових конструкцій мови (*синтаксис)* та правила тлумачення мовних конструкцій (*семантика*).

Класифікація мов програмування:

- Спочатку (у 40-х роках XX ст.) були створені мови низького рівня. ***Мови програмування низького рівня*** базуються на командах-кодах для роботи з адресами оперативної пам'яті й регістрами процесора, їх використовують для розробки швидкодіючих програм. До мов програмування низького рівня належать мови асемблера, автокоди.
- Мови високого рівня створено в 50-х роках минулого століття з метою наблизити користувача до комп'ютера. ***Мова програмування високого рівня*** – мова, близька до розмовних мов, оскільки дає змогу записувати команди у вигляді речень (складається з операторів, схожих на звичайні слова).

  Класифікація мов програмування високого рівня:
  - універсальні – мови, що призначені для написання програм для розв’язання різних типів задач із різних галузей знань (Паскаль, Сі);
  - спеціальні – мови, що призначені для написання програм для розв´язання спеціальних задач або задач однієї галузі знань (Logo).

Приклади мов програмування високого рівня: Фортран, Бейсик, Паскаль, Сі (мови процедурного типу); Пролог (мова логічного програмування); Visual Basic, Delphi (середовища візуального програмування).

Існують різні види програмування: ***структурне, логічне, об´єктно-орієнтоване, функціональне.***

***Структурне програмування*** – це процес побудови алгоритмів та програм, що виконується в такій послідовності:

1. Попередній аналіз задачі з метою розбити її на окремі прості частини (модулі). Для цього спочатку складають загальну схему алгоритму, а потім її деталізують.
2. Послідовна деталізація частин і складання програм для кожного з модулів. Виділяють основну частину і частини нижнього рівня. Кожну частину

розбивають окремо: спочатку частини верхнього рівня, а потім – нижнього. Наприкінці частини з'єднують між собою.

Мова Паскаль є структурованою мовою програмування, оскільки вона має такі властивості:

1) використання трьох базових структур алгоритмів: (слідування, розгалуження, цикл) під час роботи з кожним модулем;
2) виділення допоміжних програм (підпрограм);
3) коментування текстів програм;
4) мінімальне використання операторів безумовного переходу, що ускладнюють читання програми;
5) передбачається система тестів для перевірки правильності програми.

До основних принципів ***об'єктно-орієнтованих мов*** програмування, які відрізняють їх від процедурних, слід віднести:

- відокремлення елементів (об'єктів) програми, які пов'язані з інтерфейсом користувача, від її алгоритмічної частини;
- швидкість і простота створення, модернізація інтерфейсу програм, в якому використовуються готові елементи (блоки), що реалізують деякі великі функції (процедури взаємодії з користувачем);
- використання вже існуючих кодів, написаних іншими мовами програмування.

**Система програмування** складається з мови програмування і середовища програмування.

**Середовище програмування** — частина базового програмного забезпечення, яка підтримує процес програмування на ЕОМ.

**Середовище програмування** — це програма, що має засоби автоматизації процесів підготовки та виконання програм користувача, а саме:

- редактор текстів програм;
- довідково-інформаційну систему про мову програмування та середовище;
- бібліотеки з корисними процедурами і функціями;
- компілятор чи інтерпретатор, що не лише констатує факт помилки в програмі, а й вказує на тип і місце помилки, а в деяких випадках пропонує шляхи усунення помилок (або усуває їх);
- засоби виконання усієї програми або крок за кроком з метою виявлення семантичних помилок шляхом відстеження значень величин, які є в програмі, тощо.

Завдяки цим можливостям середовища програмування часто називають інтегрованими, або дружніми, середовищами. Приклади середовищ програмування: GW Turbo Pascal 7.0, Borland Pascal for Windows, Delphi 7, Visual Basic 5, Visual Basic 6.

**Транслятор** – це спеціальна програма, яку використовують для перекладу програм користувача, написаних мовою програмування високого рівня, у так звані машинні коди, зрозумілі процесору.

Транслятори створюють для кожної мови програмування. Отже, транслятори перекладають команди користувача в набір команд процесора. Це дає змогу використовувати програми, створені мовою високого рівня, на різних типах машин. Іншими словами, транслятор — це перекладач.

Транслятори бувають двох типів: інтерпретатори і компілятори.

**Інтерпретатор** перекладає команди програми в машинні коди і відразу їх виконує. Повторний запуск програми на виконання супроводжується повторним перекладом. Це невигідно, якщо програми великі. Приклади інтерпретаторів: інтерпретатори мов GWBasic, QBasic.

Інтерпретатор може бути двопрохідним. Перший раз він опрацьовує кожну команду відразу після її введення, аналізує на наявність помилок, але не виконує її. Другий «прохід» після запуску програми на виконання: команди одна за одною перекладаються в машинні коди і виконуються.

**Компілятор** аналізує команди програми на наявність помилок і перекладає її в машинні коди, утворюючи так званий exe-файл програми, але не виконує його.

Exe-файл може тимчасово зберігатися в оперативній пам'яті або постійно – на диску. В останньому випадку програму, точніше створений exe-файл, можна запускати на виконання багаторазово. Це зручно для великих програм, бо заощаджується час на перекладання програми в машинні коди. Крім цього, компілятор «стискує» exe-файл (тому його не варто архівувати з метою економії місця на диску). В exe-файли не можна вносити зміни без спеціальних програмних засобів. Зазвичай всі прикладні програми поширюють у вигляді exe-файлів. Приклади компіляторів: Turbo Basic, Quick Basic 4.5, Turbo Pascal 7.0 тощо.

Отже, компілятор дає змогу, переклавши програму в машинні коди лише один раз, отримати exe-файл, придатний для багаторазового виконання.

## Лекція. Ознайомлення з середовищем програмування Turbo Pascal. Редагування тексту

**Мета, завдання лекції:** ознайомити студентів з можливостями,інтерфейсом та основними прийомами роботи в середовищі програмування Turbo Pascal.

**План і організаційна структура лекції:**

1. Запуск програм на виконання.
2. Інтерфейс програми Turbo Pascal.
3. Головні етапи роботи користувача.

**Зміст лекційного матеріалу:**

Щоб увійти в середовище Turbo Pascal, потрібно:

1. Увійти в каталог системи програмування. (Наприклад: TP 7.0).
2. Запустити командний файл turbo.exe.

На екрані з'явиться головне меню системи.

У верхньому рядку подані всі команди головного меню, а в нижньому – рядок статусу, в якому перелічені імена функціональних клавіш, призначених для виконання деяких операцій в конкретній ситуації.

Щоб перейти в головне меню, потрібно натиснути клавішу **F10.** Щоб вийти з головного меню і повернутись в редактор тексту – натиснути клавішу **Esc.**

***Пункти меню:***

**File** – для роботи з файлами;
**Edit –** для редагування файлу;
**Search** – для шукання чи заміни заданого тексту;
**Run** – для виконання програми;
**Compile** – для компіляції та створення exe-файлу;
**Debug** – для налагодження програми;
**Options** – для конфігурування середовища;
**Window** – для конфігурування вікон та роботи з ними;
**Help** – для отримання допомоги.

Виконати команду означає: з допомогою клавіш управління курсором виставити курсор на потрібну команду і натиснути клавішу **Enter.** (Виконати команду можна ще іншими способами: з допомогою маніпулятора "миша" або натисканням комбінацій клавіш **Alt** + та буква, яка виділена в назві команди).

Кожна команда головного меню має один або декілька (вкладених одне в одне) підменю.

**Головні етапи роботи користувача:**

1. Спочатку активізують головне меню (для цього натискають на **F10**) і пункт **File**. Отримують додаткове спадне меню. У ньому вибирають команду **New**. Середовище переходить у режим створення нового файлу з назвою NONAME00.PAS. Набирають текст деякої програми.
2. Для виправлення очевидних помилок уведення користуються традиційними прийомами редагування тексту, зокрема, такими клавішами чи їхніми комбінаціями для роботи з блоками тексту:

**Enter** – перейти на новий рядок, вставити порожній рядок чи розбити рядок на два рядки;

**Delete** - вилучити символ над курсором;

**Insert** - перемкнути режими вставки чи заміни;

**Shift+стрілки** - виокремити чи зняти виокремлення тексту;

**Ctrl+Insert** - копіювати блок у буфер;

**Shift+Del** - перемістити блок з тексту в буфер;

**Shift+Insert** - вставити текст з буфера у позначене курсором місце основного тексту;

**Ctrl+Del** - вилучити виокремлений блок з тексту;

**Ctrl+Y** - вилучити вибраний рядок з тексту.

3. Якщо очевидних помилок немає, програму можна виконати за допомогою пункту головного меню **Run** або комбінації клавіш **Ctrl+F9**. Відбудеться компіляція і виконання програми. Щоб виконати компіляцію програми без її виконання, натискають на **Alt+F9**.
4. Якщо система виявить синтаксичні помилки, то про це буде негайно повідомлено. Курсор буде в рядку, де допущено помилку, або безпосередньо показуватиме на позицію з помилкою. У верхньому рядку екрана буде повідомлення червоного кольору про зміст помилки, що суттєво полегшує її виправлення. Середовище перебуватиме в режимі редагування, і помилку легко можна виправити. Програма найчастіше фіксує такі помилки початківців (у дужках наведено текст повідомлення):
   а) пропущено крапку з комою (*Error 85: ";" expected*);
   б) немає дужок *(Error 89: ")" expected або Ezzog 5: Syntax*);
   в) не описана змінна (*Error 3: Unknown identifier*);
   г) повторне використання імен (*Error 4: Dublicate identifier*);
   д) неузгодження типів (*Error 26: Type mismatch*);
   е) потрібна кома (*Error* 87: "," *expected*) тощо. Помилку виправляють. Удруге виконують програму (натисканням на Ctrl+F9).
5. Якщо синтаксичних помилок немає, програма буде виконана. Результати (якщо ви про них подбали) можна побачити у вікні результатів, для чого натискають на **Alt+F5** або використовують команди меню **Debug, Screen Server**. Натиснувши після перегляду результатів на будь-яку клавішу, переходять у режим редагування програми.
6. Щоб для заданого раз-файлу створити exe-файл, активізують пункт **Compile**. Вибирають **Destination**. Натисканням на клавішу вводу вибирають режим **Disc.** Натискають на **Alt+F9**, і в поточний каталог на диску буде записано exe-файл, який можна виконувати поза середовищем.
7. Щоб зберегти текст програми у файлі з розширенням pas, знову активізують пункт **File** і підпункт **Save As...**, якщо файлу дають нове ім'я, або підпункт **Save** — для зберігання файлу зі старим іменем. Щоб цю операцію виконати швидко, достатньо натиснути на **клавішу F2.**
8. Для закінчення сеансу роботи і виходу із середовища потрібно ввійти в головне меню (**F10**), у підпункт **File** (**Alt+F**) і вибрати команду **Exit** (натискають на літеру **x**). Швидкий вихід забезпечує комбінація клавіш **Alt+x.**
9. Для відкривання файлу з текстом програми, що є на диску, входять у головне меню (**F10**), вибирають пункт **File** і команду **Open**. Щоб цю дію зробити швидко,

достатньо натиснути на клавішу **F3**. У вікні, що з'явиться на екрані, набирають назву файлу і натискають на клавішу вводу. Однак ліпше зробити так: за допомогою клавіші **Tab** перейти в нижнє вікно, вибрати з меню імен файлів потрібний файл і натиснути на клавішу вводу. Текст програми буде занесено у вікно редагування.

**10.** Вікон з програмами може бути декілька. Переходити від одної програми до іншої можна за допомогою клавіші **F6** або **Alt+0** (нуль). Щоб розкрити на весь екран чи згорнути вікно, користуються клавішею **F5**. Зручно розташувати вікна на екрані можна засобами пункту **Window.** Щоб закрити вікно, натискають на **Alt+F3** або клацають мишею на значку прямокутника у верхньому лівому куті рамки вікна.

**11.** Якщо унаслідок неправильних дій користувача вікно стало неактивним (зникла подвійна рамка навколо вікна), то натисніть на клавішу **Esc**.

**12. F1 –** довідка.

Текстовий редактор версії Turbo Pascal 7.0 дозволяє обробляти файли довжиною до 1Мб.

## Практична робота. Робота в середовищі програмування. Запуск програм на виконання

**Мета:** ознайомлення з середовищем програмування Паскаль, формування вмінь редагувати, записувати та виконувати програми.

**Завдання:**

- *студенти повинні знати:* інтерфейс, призначення меню середовищ програмування Turbo Pascal та Pascal ABC;
- *студенти повинні вміти:* запускати середовище програмування, створювати та зберігати файл програми, запускати програми на виконання, переглядати результат, виконувати компіляцію, виявляти та виправляти помилки.

**Обладнання:** середовище програмування.

**Хід виконання:**

**1.** Запустіть програму з середовищем мови програмування Turbo Pascal.

**2.** Створіть новий файл.

Натисніть клавішу **F10**, щоб активізувати меню. Виберіть клавішами-стрілками (або клацніть мишею) команду **File**, а в ній **New**. Отримаєте вікно у подвійній рамці з назвою NONAME00.PAS. Це вікно називається вікном редагування тексту програми.

**3.** Наберіть у цьому вікні текст ( кожне слово у новому рядку): своє ім'я і прізвище та адресу..

**4.** Запишіть утворений файл на диск із назвою F1*.pas.

**File\Save as...** В отриманому діалоговому вікні введіть назву файлу **F1*.pas** і клацніть мишею на **ОК** або, вибравши клавішею **Tab** кнопку **ОК,** натисніть на клавішу вводу.

**5.** Знову розгляньте вікно редактора тексту.

Введіть порожні рядки між словами. Для цього після кожного слова натисніть на клавішу **ENTER**.

**6.** З'єднайте ім'я і прізвище в один рядок.

**7.** Розмістіть курсор посередині прізвища і введіть три будь-які літери, наприклад 'а'.

Режими перемикаються клавішею **Insert.**

**8.** Адресу повторіть чотири рази методом копіювання і використовуючи буфер обміну.

Для цього потрібно виділити потрібний текст мишею або клавішами **Shift+стрілки** і використовувати команди меню **Edit\Copy (**копіює в буфер обміну) **Edit\Paste** (вставляє з буферу обміну).

**9.** Збережіть внесені зміни у файл F1*.pas.

**10.** Продемонструйте текст файлу викладачу і вийдіть з програми за допомогою команд **File\Exit** або комбінації клавіш **Alt+X.**

**11.** Завантажте середовище програмування Turbo Pascal.

**12.** Введіть з клавіатури текст програми:

***Для хлопців:***

**program** *ideal;*
**var** *R,V: integer;*

```
        M:real;
    begin
        writeln ('Введіть Ваш ріст:');
        read (R);
      writeln ('Введіть скільки Вам років:');
        read (V);
        M:=(3*R-450+V)*0.25+45;
        writeln ('Ваша ідеальна маса:', M:2:2, 'кг');
    end.
```

***Для дівчат:***

```
    program ideal;
    var R,V: integer;
        M:real;
    begin
        writeln ('Введіть Ваш ріст:');
        read (R);
      writeln ('Введіть скільки Вам років:');
        read (V);
        M:=(3*R-450+V)*0.225+40.5;
        writeln ('Ваша ідеальна маса:', M:2:2, 'кг');
    end.
```

**13.** Запустіть програму на виконання.

**14.** Перегляньте результати виконання даної програми.

**15.** Запишіть утворений файл на диск із назвою F2*.pas.

**16.** Створіть виконуваний .exe файл з тим самим іменем.

**17.** Вийдіть із середовища програмування.

**18.** Завантажте щойно створений .exe файл.

**19.** Запустіть середовище програмування Pascal ABC.

**20.** Створіть аналогічну програму в цьому середовищі та збережіть файл на диску з іменем Прізвище. pas.

**21.** Запустіть файл на компіляцію та виконайте його.

**22.** Робота в якому середовищі програмування є для вас простішою? Чому?

**23.** Продемонструйте отримані результати викладачу.

## Лекція. Мова програмування Паскаль: алфавіт, синтаксис

**Мета, завдання лекції:** ознайомити студентів з алфавітом мови програмування Паскаль, типами даних, структурою програми.

**План і організаційна структура лекції:**

1. Алфавіт, службові слова й імена об'єктів.
2. Структура паскаль-програми. Команда присвоєння.
3. Знайомство з паскаль-програмою і поняттям «змінна».
4. Арифметичні вирази.
5. Функції.

**Зміст лекційного матеріалу:**

***Алфавіт, службові слова й імена об'єктів***

У природних мовах тексти будують так:

Алфавіт мови ⇨ Слова ⇨ Речення ⇨ Текст.

В алгоритмічних мовах простежується цілковита аналогія:

Алфавіт ⇨ Слова ⇨ Команди ⇨ Програма.

***Алфавіт мови програмування*** містить майже всі (за деякими винятками) символи, що є на клавіатурі:

- латинські символи (великі та малі);
- символи кирилиці (великі та малі);
- цифри від 0 до 9;
- математичні символи (+, -, *, /, =, <, >);
- розділові знаки (кома, крапка, двокрапка, крапка з комою, пропуск, лапки, квадратні, круглі, фігурні дужки) та ін.

**Слова поділяються на службові, ідентифікатори та стандартні імена.**

***Службові слова***

Службові слова призначені для написання команд, їх є невелика кількість. Розглянемо основні службові слова мови Паскаль (запам'ятовувати їх не треба):

**and** – і
**array** – масив
**begin** – початок
**case** – вибір
**const —** сталі
**div** – ділення без остачі
**do** – виконати
**downto** – униз до
**else** – інакше
**end** – кінець
**file** – файл
**for** – для
**function** – функція
**goto** – перейти до
**if** – якщо
**label** – позначка
**mod** – остача
**not** – не
**of** – з
**or** – або
**procedure** – процедура
**program** – програма
**record** – запис
**repeat** – повторювати
**then** – то
**while** – доки.

***Імена, які утворює користувач (ідентифікатори)***

Складаючи програму, користувач описує різні об'єкти і надає їм імена на свій розсуд. Тут простежується аналогія з математикою та фізикою, де різні величини позначають різними буквами, наприклад: a, b, c – довжини сторін трикутника, H – висота, S – шлях чи площа. В алгоритмічних мовах дане, яке міститиме інформацію про висоту, можна назвати різними способами: h, hight чи інакше. Придумуючи імена, треба дотримуватися певних правил.

***Правила утворення імен користувачем:***

- ім'я може складатися лише з латинських літер, цифр і символа «_» (риска знизу);
- ім'я не може бути службовим словом;
- цифра не може бути першим символом в імені;
- літери можуть бути малими або великими;
- бажано, щоб імена були короткими (до 60 символів) і відповідали суті об'єкта;
- пропуски в іменах не допускаються;
- два різні об'єкти не можна позначати одним іменем.

В іменах та службових словах великі та малі букви рівноправні: імена A та a (або MyName та myname) означають одне й те ж.

На відміну від математики та фізики в інформатиці можна використовувати довгі імена, наприклад, Hight, Myname, Mynumber, My_program тощо.

*Приклади правильно утворених імен*: a, b, c, x, z, a1, a2,..., a100, alpha, cat, My_number.

*Неправильно утворені імена*: 10a, 11b, a+2, a.

***Стандартні імена*** діляться на декілька груп:

- назви стандартних типів даних: ***integer*** (цілий), ***real***(дійсний), ***boolean*** (логічний), ***char*** (символьний), ***text*** (текстовий файл) тощо;
- назви стандартних сталих: ***false*** (хибність), ***true*** (істинність), ***maxint*** (максимальне ціле число), ***pi*** (число пі) тощо;
- назви стандартних функцій: ***abs***(x) - модуль числа x, ***arctan***(x), ***cos***(x), ***exp***(x), ***ln***(x), ***sin***(x), ***sqr***(x) – піднесення числа x до квадрату, ***sqrt***(x) - квадратний корінь з x;
- назви команд для введення і виведення даних: ***read, readln, write, writeln*** тощо.

***Знайомство з паскаль-програмою і поняттям «змінна»***

Програма складається з команд. Команди призначені для описування і перетворення даних. Команди відокремлюють одну від одної крапкою з комою (;).

*Приклад 1*. Розглянемо програму, що виводить на екран фразу «Хочу мати з інформатики 12».

```
program First;
var a : integer;
begin
  a:= 12;
```

```
    writeln ('Хочу мати з інформатики ', a)
end.
```

Службові слова – **program**, **var**, **begin**, **end**

Стандартні імена – integer, writeln.

Імена користувача – First (ім'я програми), a.

Ім'я a позначає об'єкт, який називається змінною і якому командою присвоєння надано значення 12.

У мові Паскаль усі змінні обов'язково потрібно оголосити на початку програми у розділі оголошення змінних **var.**

***Структура паскаль-програми. Команда присвоєння***

Програма складається з описової частини – заголовка, розділів описів та оголошень, та виконуваної – розділу команд:

```
program <Назва програми>;
   <Розділи описів та оголошень>
begin
  <Розділ команд>
end.
```

В розділі описів та оголошень описують дані, які будуть використовуватись в задачі.

Під даним розуміють об'єкт – порцію інформації, що зберігається в пам'яті комп'ютера, має значення з деякої множини допустимих значень, над якими визначені допустимі операції.

Дані бувають сталі та змінні.

*Стале* дане не може змінити свого значення під час виконання програми. Прикладами сталих цілих даних є числа: 5, -10.

Змінні призначені для зберігання конкретних значень в оперативній пам'яті під час виконання програми.

Змінна може набувати різних значень. *Фізичний зміст змінної:* змінна – це ділянка оперативної пам'яті, куди комп'ютер записує або звідки читає дане.

Значення змінній надають командою присвоєння чи командою введення даних.

*Приклад 2.* Розглянемо такі математичні та фізичні формули:

y = 2x – рівняння прямої (лінійна функція);

p = 2(a+в) – периметр прямокутника;

s=v*t – шлях у рівномірному русі;

$h=gt^2$ – шлях вільного падіння.

Наведені формули в деякій програмі записують за допомогою команд присвоєння так:

1) y:=2*x;  2) p:=2*(a+b);  3) s:=v*t;  4) h:=g*t*t/2.

Символ множення «*» потрібно записувати обов'язково.

***Розділ оголошення змінних***

Розв'язуючи задачу, користувач має проаналізувати, скільки змінних треба і до якого типу їх віднести. Змінні потрібно оголосити на початку програми у розділі оголошення змінних **var**, який має такий вигляд:

**var**

<перший список імен змінних> : <назва типу 1>;

. . .

<n-й список імен змінних> : <назва типу n>;

***Розділ оголошення сталих***

Якщо значення даних не змінюватимуться під час виконання програми, то їх оголошують як сталі (константи) у розділі оголошення сталих:

**cons**t <стала1 >=<значення1 >; <стала2>=<значення2>; і т.д.

Н-д: const suma=5; f1=6;

*Дані цілого типу*

| *Назва типу* | *Пояснення* | *Обсяг* | *Діапазон* |
|---|---|---|---|
| byte | цілі дуже короткі | 1 байт | 0 - 255 |
| integer | цілі короткі | 2 байти | від -32 768 до 32 767 |
| longint | цілі довгі | 4 байти | від -2 147 483 648 до 2 147 483 647 |

Елементи списку відокремлюють комою, а команди – крапкою з комою.

*Приклад 3.* Від міста А до міста В автомобіль їхав $t_1 = 5$ год із середньою швидкістю $v_1 = 70$ км/год, від В до С – $t_2 = 4$ год зі швидкістю $v_2 = 75$ км/год. Визначити відстань між містами.

```
program Vidstan;
var t1, t2, v1, v2, ab, bc, s : integer;
begin
    ab:=v1*t1; bc:=v2*t2;
    s:=ab+bc;
    writeln (ab:6,bc:6,s:6);
    readln
end.
```

Виконаємо програму і на екрані отримаємо: 350 300 650.

*Завдання 1*. Модифікуйте програму на випадок чотирьох міст.

*Завдання 2*. Скільки секунд мають доба, тиждень, рік?

*Завдання 3*. Швидкість світла 299 792 км/с. Яку відстань долає світло за хвилину, годину, добу?

*Дані дійсного типу*

Якщо числові дані не цілі числа або виходять за межі типу longint, треба застосувати дійсні типи даних **real.**

В інформатиці число належить до дійсного типу, якщо в його зображенні є десяткова крапка (12.5, 5.0) або якщо воно записане у показниковій формі, наприклад, 12.5E0, 5.0E0.

*Правила написання чисел:*

Цілу і дробову частини у десятковому числі розділяє крапка.

Знак «+» перед додатним числом можна не писати.

Показниковий вигляд дійсного числа: **mEp = m*10^p**

Приклад:

6.25E+01 = 6.25 • $10^1$ = 62.5;

-0.125E-01= -0.125 • $10^{-1}$= -0.0125

| *Назва типу* | *Пояснення* | *Обсяг* | *Діапазон* |
|---|---|---|---|
| single | дійсні короткі | 4 байти | |
| real | дійсні | 6 байтів | 2,9E-39 – 1,7E+38 |
| double | дійсні довгі | 8 байтів | |
| extended | дійсні дуже довгі | 12 байтів | $10^{308}$ |

*Дані типу String*

Дані, значеннями яких є група символів (слово або деякий текст), називаються текстовими (інший термін – рядки). Назва цього типу даних – *string.* Ознакою текстової сталої є одинарні лапки; (апострофи), між якими записана група символів, а саме: '5', 'Сума=', 'Оля Курочка'.

Отже, 2005 – це ціла числова стала, а '2005' – текстова стала. Якщо текст містить апостроф, то він дублюється, наприклад, 'Мар"яна', 'ім"я'.

Текстові дані типу *string* можуть містити до 255 символів, однак часто потрібна менша кількість символів п, яку задають в описах так: *string* [n].

*Приклад 4.* Оголосити змінні a1, a2, a3 як дійсні, b1, b2 - як цілі, а c1 – як текстову можна так:

**var** a1, a2, a3 : real; b1, b2 : integer; c1 : string; .

*Приклад 5.* Туристи придбали на потяг п'ять квитків для дорослих за 18,50 грн кожний і три квитки для дітей за 9,25 грн. Скласти програму для визначення суми, яку треба заплатити за всі квитки, і вивести на екран повідомлення «Щасливої дороги!».

Розглянемо такі сталі: k1– кількість квитків для дорослих, k2 – кількість квитків для дітей (тип integer); c1 – ціна квитка для дорослого, c2 – ціна квитка для дитини, а також змінні: suma – загальна сума до оплати (тип real); p – повідомлення (тип string).

```
program Zad1;
const k1 = 5; c1 = 18.50; k2 = 3; c2 = 9.25;
var suma : real; p : string;
begin p := 'Щасливої дороги!';
      suma := k1 * c1 + k2 * c2;
      writeln ('Сума = ' , suma:6:2);
      writeln(p)
end.
```

Запис suma: 6: 2 означає, що для виведення результату на екран система надасть шість позицій, з них дві позиції для цифр після десяткової крапки.

*Використання стандартної сталої pi*

Для позначення числа p (3.1415...) у мові Паскаль є стандартна стала, яка має ім'я **pi**. Це ім'я можна використовувати у командах присвоєння, наприклад, площу круга з з радіусом r визначають так: s := pi * r * r. Можна також писати s := 3.14 * r * r, але у цьому випадку результат буде менш точним.

***Розділ команд***

Цей розділ містить команди, призначені для перетворення даних. Команди прийнято записувати одну під одною, роблячи пропуски між словами і відступи від лівого краю для наочності. Команди відокремлюють символом «;». Короткі команди можна розміщувати в одному рядку. Одну довгу команду можна записувати у декількох рядках, не розриваючи слів.

Після слова **begin** та перед **end** символ «;» можна не писати. Програма закінчується крапкою.

Команда присвоєння призначена для надання значення змінній. Цю дію позначають двома символами **:=.**

Наприклад, a := 12. Загальний вигляд команди присвоєння такий:

***<ім'я змінної> := <вираз>***

Два символи (:=) читаємо як «присвоїти», «надати».

***Арифметичні вирази***

**Вираз** – це запис мовою програмування правої частини деякої формули, призначеної для перетворення даних.

*Дія команди.* Обчислюється вираз, і результат присвоюється зазначеній змінній.

За допомогою команди присвоєння і виразів записують традиційні для математики і фізики формули s=vt, p=F/S) тощо.

Арифметичний вираз містити числа, змінні, функції, з'єднані символами арифметичних операцій:

| *Операція* | *Приклад* | *Пріоритет* |
|---|---|---|
| * множення | 5 * 2 = 10 | 1 |
| / ділення | 4 / 2 = 2.0 | 1 |
| + додавання | 5 + 2 = 7 | 2 |
| - віднімання | 5-2 = 3 | 2 |

Результатом ділення двох цілих чисел завжди є дійсне число.

Операції **div** і **mod**.

| *Операція* | *Приклад* | *Пріоритет* |
|---|---|---|
| **div** – ціла частина від ділення | 5 **div** 2 = 2 | 1 |
| **mod** – остача від ділення | 5 **mod** 2 = 1 | 1 |

*Приклад 6.* Нехай змінна a типу integer має значення 5. Обчислити вирази:

а) a + 2*a — 6/3;

б) (20- 2 *a)/ a - 1;

в) 10 / a * 2;

г) 10 div a * 2;

д) 5 + 9 mod a * 3.

*Розв'язування:*

а) 5 + 2*5 - 6/3=5 + 10 - 2.0 = 13.0;

б) (20 - 10) / 5 - 1 = 2.0 - 1 = 1.0;

в) 10 / 5 * 2 = 2.0 * 2 = 4.0; д) 5 + 9 mod 5*3 = 5 + 4*3 = 5 +12 = 17.

г) 10 div 5 * 2 = 2 * 2 = 4;

***Функції***

| *Математичні позначення* | *Написання мовою Паскаль* | *Пояснення* |
|---|---|---|
| **Sin x** | sin(x) | |
| **Cos x** | cos(x) | x задають радіанами |
| **Tg x** | sin(x)/cos(x) | |
| **Arctg x** | arctan(x) | Арктангенс |
| **IxI** | abs(x) | Абсолютна величина |
| $\sqrt{x}$ | sqrtx) | Квадратний корінь |
| **Ln x** | ln(x) | Натуральний логарифм |
| $\mathbf{e^x}$ | exp(x) | Експонента |
| $\mathbf{x^2}$ | sqr(x) | Піднесення до квадрата |
| | random(x) | Випадкове ціле число з проміжку [0; x) |

**trunc**(<вираз>) - відкидає дробову частину числа;

**round** (<вираз>) - заокруглює число до найближчого цілого.

Наприклад, trunc (3.6) і round (3.4) дає 3, а round (3.6) дає 4.

У мові Паскаль немає операції піднесення до степеня, її реалізують або через операцію множення, наприклад, $x^3$ = x*x*x, або, якщо показник степеня – велике або дробове число, за допомогою основної логарифмічної тотожності $\mathbf{x^a = e^{alnx}}$.
Наприклад, $x^5$, де x >0, записують так: exp(5 * ln(x)).

***Правила утворення та обчислення виразів***

1. *Правило пріоритетів операцій:*

Спочатку виконуються операції вищого пріоритету. Операції однакового пріоритету виконуються послідовно зліва направо.

2. *Правило дужок:*

Порядок виконання операцій можна змінити за допомогою круглих дужок.

3. *Правило лінійного запису:*

Чисельники і знаменники дробів та індекси записують в одну лінію.

4. *Правило коректних імен:*

Замість грецьких чи українських літер в іменах величин (змінних) потрібно писати латинські (літеру чи слово).

| *Математичні вирази* | *Арифметичні вирази* | |
|---|---|---|
| | *Правильно* | *Неправильно* |
| $\frac{a}{bc}$ | a/b/c або a/(b*c) | a/b*c |
| 5a+3b | 5*a+3*b | 5a+3b |

| | | |
|---|---|---|
| $\frac{5x}{\cos 5x + b}$ | 5*x/(cos(5*x)+b) | 5*x/(cos(5x)+b |

## Практична робота. Опис програм за правилами мови програмування

**Мета:** ознайомлення з середовищем програмування Паскаль, формування вмінь редагувати, записувати та виконувати програми.

**Завдання:**

- *студенти повинні знати:* інтерфейс, призначення меню середовища програмування Turbo Pascal, структуру програми, синтаксис операторів введення-виведення;
- *студенти повинні вміти:* запускати середовище програмування, запускати програми на виконання, переглядати результат, зберігати програму, складати та виконувати в середовищі найпростіші програми на введення-виведення даних.

**Обладнання:** середовище програмування.

**Хід виконання:**

***Завдання 1.*** Розгляньте приклад розв'язання задачі та реалізуйте його в середовищі програмування.

Задати два цілі числа, наприклад, 24 і 5. Обчислити і вивести на екран їхню суму, різницю і добуток.

Для розв'язування задачі складемо програму мовою Паскаль:

```
program Zadacha1;
var a, b, s, r, d. : integer;
begin
    a := 24;  b := 5;
    s := a + b;
    r := a - b;
    d:=a*b;
    writeln (s, r, d)
end.
```

*Вказівка до виконання:* Уведіть текст програми і виконайте команду **Run** з головного меню або натисніть на комбінацію **CTRL+F9**. Якщо будуть помилки, виправте їх. Ще раз дайте на виконання команду **Run**. Натиснувши клавіші **Alt+F5**, отримаємо на екрані результат: 2919120.

Зверніть увагу, що числа на екрані зливаються і результат зрозуміти важко. Щоб числа не зливалися, команду writeln (s, r, d) замініть на writeln (s:4, r:4, d:6). Тут запис s:4 означає не ділення (ділення позначають /), а те, що цілому числу треба надати на екрані чотири позиції. Виконайте програму ще раз. На екрані побачите:
29 19  120.

***Завдання 2***. Введіть і реалізуйте зразок програми обчислення значення функції при значеннях a=6, b=14, c=4:

Y=(a+b-c)/(a-b) при a=6. b=14, c=4

```
program Probota2;
var
   Y: real;
```

```
    a, b, c: integer;
begin
      write(' Введіть значення a:  ');
      readln(a);
      write(' Введіть значення b:  ');
      readln(b);
      write(' Введіть значення c:  ');
      readln(c);
      Y:=(a+b-c)/(a-b);
      writeln (' Значення функції Y=',Y);
end.
```

***Завдання 3.*** Реалізуйте розв'язання задачі в середовищі програмування.

Квіткова клумба має форму круга. Обчисліть її периметр і площу за заданим радіусом.

***Завдання 4.*** Реалізуйте розв'язання задачі в середовищі програмування.

Обчисліть площу бічної поверхні $S=2\pi rh$ та об'єм $v=\pi r^2 h$ діжки за заданими висотою h і радіусом основи r.

***Контрольні запитання:***

1) Мова програмування Паскаль: алфавіт, синтаксис.
2) Опис програм за правилами мови програмування. Структура паскаль-програми.
3) Типи даних в Паскаль.
4) Використання функцій при записі виразів в Паскаль.
5) Поняття оператора: оператори введення-виведення, присвоєння.

## Лекція. Поняття оператора: оператори введення-виведення, присвоєння

**Мета, завдання лекції:** ознайомити студентів з операторами введення, виведення, присвоєння.

**План і організаційна структура лекції:**

1. Оператор присвоєння.
2. Оператори введення Read, Readln.
3. Оператори виведення Write, Writeln.

**Зміст лекційного матеріалу:**

Значення змінній надають командою присвоєння чи командою введення даних.

*Приклад1.* Розглянемо такі математичні та фізичні формули:

y = 2x – рівняння прямої (лінійна функція);

p = 2(a+в) – периметр прямокутника;

s = v*t – шлях у рівномірному русі;

$h = gt^2$ – шлях вільного падіння.

Наведені формули в деякій програмі записують за допомогою команд присвоєння так:

1) y:=2*x; 2) p:=2*(a+b); 3) s: = v*t; 4) h:=g*t*t/2.

Символ множення «*» потрібно записувати обов'язково.

***Введення даних***

Команда введення даних має вигляд

**read** (<список змінних>)

*Дія команди.* Виконання програми призупиняється для введення значень змінних. Якщо набрати недостатню кількість чисел, то програма очікуватиме на наступні дані. Крім чисел, цією командою можна вводити текстові дані типу string (тексти набирають без лапок).

*Зауваження.* Використовують також різновид команди введення **readln** (<список змінних>). Середовище ігноруватиме дані, якщо їх набрано у рядку більше, ніж є змінних у списку. Команду **readln** без списку змінних використовують для створення паузи з метою перегляду результатів на екрані. Паузу закінчують натисканням будь-якої клавіші на клавіатурі.

***Виведення даних***

Команда виведення призначена для виведення значень на екран. Її загальний вигляд такий:

**write** (<список>)

Список може складатися зі сталих, змінних, виразів, а також записаних у лапках текстів.

*Дія команди.* Вирази обчислюються та їхні значення виводяться на екран без пропусків. Це може призвести до злиття даних на екрані.

Наступна команда **write** виводитиме дані у тому ж рядку. Щоб виводити дані у наступному рядку, застосовують команду **writeln.**

*Приклад 2.* Нехай змінні a, b та c отримали такі значення: 2, 5, 1.

Команда write (a, 9, b+c) виведе у лівому кутку екрана таке: 296.

*Задача*. Скласти програму для виведення на екран адреси школи чи гімназії.

```
program Address;
uses Crt;
begin clrscr;
      writeln(' 01011 м. Київ ');
      writeln (' вул. Південна, 3');
      writeln (' Школа чи гімназія № чи назва');
end.
```

***Формати виведення***

Формат :**n** надає на екрані n позицій для зображення цілого числа, а також тексту.

Формат :n:k надає n позицій для дійсного числа, з них k - для цифр після десяткової крапки. Якщо позицій забагато, то перед цілою частиною числа будуть пропуски. Якщо замало позицій для дробової частини, то відбувається заокруглення числа. Якщо замало позицій для цілої частини, то компілятор додасть позиції. Знак «-» і десяткова крапка входять до кількості позицій n.

*Приклад.* Розглянемо команди виведення чисел та їхній вигляд на екрані.

| Команди | Вигляд чисел на екрані |
|---|---|
| write (5, 15, 25, -35) | 51525-35 |
| write (5:2, 15:3, 25:4, -35:4) | _5_15__25_-35 |
| write (6+2:2, +50:4). | _8_50 |
| write (2.5:7:2) | ___2.5 |
| write (-2.5:6:2, 3.548:6:2) | _-2.50_3.55 |

***Імітація діалогів***

Діалоговий алгоритм імітує діалог між користувачем і комп'ютером. Відповідна програма складається в основному з команд writeln та readln. Діалог використовують, зокрема, під час уведення даних з метою отримати на екрані підказку про те, що саме треба ввести, наприклад так:

```
write ('Введіть значення радіуса К: '); readln (К);
```

*Задача.* Скласти програму діалогу користувача з комп'ютером за таким сценарієм: комп'ютер запитує користувача, як його звати, користувач вводить своє ім'я, комп'ютер вітається і пропонує з ним поспілкуватися на тему улюбленого предмета.

Програма розв'язку даної задачі мовою програмування Паскаль:

```
  Program PROBOTA3;
     Var
        STl,ST2,ST3:string[10];
     begin
          writeln('Добрий день шановні учні!');
          writeln('Я — комп"ютер фірми IBM.');
          writeln('Радий познайомитися з Вами.');
          writeln ('Як Вас звати? ');
          readln(ST1);
          writeln( 'Дуже приємно. ',ST1);
          writeln('Я думаю що ми будемо друзями.');
```

```
        writeln('В якому класі Ви навчаєтесь? ');
        readln(ST2);
        writeln('Я знаю що в ',ST2,' класі дуже гарні учні.');
        writeln('Вам подобається Вам вивчати інформатику? ');
        readln(ST3);
        writeln('Бажаю Вам успіху у вивченні даного предмету!');
        writeln('Завжди отримуйте тільки відмінні оцінки!')
    end.
```

*Результати виконання програми:*

Добрий день, шановні учні!
Я – комп'ютер фірми IBM.
Радий познайомитися з Вами.
Як Вас звати?
Микола
Дуже приємно. Микола
Я думаю що ми будемо друзями.
В якому класі Ви навчаєтесь?
11-А
Я знаю що в 11 -А класі дуже гарні учні.
Чи подобається Вам вивчати інформатику?
Так
Бажаю Вам успіху у вивченні даного предмету!
Завжди отримуйте тільки відмінні оцінки!

***Завдання для самостійного виконання:***

1. Нехай a=2, b=4, c=7. Що буде виведено на екран після виконання команд:
   а) writeln (2, 5, 5*3, a);
   б) writeln (6:2, a+b);
   в) writeln (a*b:2, 2*c, 8);
   г) writeln ('a=', a:2,' 2*2=',4);
   д) writeln( 'Заголовок таблиці ', c)?
2. Нехай a=245, b=-435, c=35.126. Що буде виведено на екран після виконання команд:
   ) write(a,b,c);
   б) writeln (a:5,b:4);
   в) write(c:8:1);
   г) write (a:4,b:2);

   д) write(c:6:2);
   е) writeln((a-b):4,c*2:6:2);
   є) writeln('a=', a:3, 'b=', b:4, ' c=', c:3:0)?Реалізуйте такий сюжет: комп'ютер повинен опитати прізвище, ім'я, по батькові, а також рік народження особи та вивести введені користувачем дані на екран.
3. У магазині комп'ютер повинен опитати вартість покупки і суму, яку сплачує покупець. Виведіть повідомлення про решту, яку має видати продавець.

## Практична робота. Поняття оператора: оператори введення-виведення, присвоєння

**Мета:** формування навичок складання, введення і редагування найпростіших програм на введення і виведення даних.

**Завдання:**

- *студенти повинні знати:* інтерфейс, призначення меню середовища програмування Turbo Pascal, структуру програми, синтаксис операторів введення-виведення, присвоєння;
- *студенти повинні вміти:* запускати середовище програмування, запускати програми на виконання, переглядати результат, зберігати програму, складати та виконувати в середовищі найпростіші програми на введення-виведення даних.

**Обладнання:** дошка, зошит, середовище програмування.

**Хід виконання:**

1. Вивчити теоретичний матеріал з теми:
   - запис констант, змінних;
   - правила запису алгебраїчних виразів;
   - використання вказівки присвоєння;
   - організація введення і виведення даних.
2. Скласти програму знаходження:

   В-1: площі трапеції

   В-2: гіпотенузи прямокутного трикутника
3. Виконати дану програму для значень:

   В-1: a=4. b=8, h=5

   В-2: a=5, b=4
4. Зберегти створену програму.

***Додаткове завдання:***

1. Створити програми вашого варіанту. Відлагодити та виконати програму (всі програму складати у вигляді діалогу, використовуючи оператори введення-виведення даних)
2. Записати в зошити код складених програм та результат обчислень.

| Вар. № завд. | I | II | III | IV |
|---|---|---|---|---|
| 1 | Користувач задає значення величин A, B. ПК розраховує суму цих чисел. | Користувач задає значення величин C, K. ПК розраховує добуток цих чисел. | Користувач задає значення величин T, X.. ПК розраховує різницю цих чисел. | Користувач задає значення величин S, F. ПК розраховує частку цих чисел. |

| | | | | |
|---|---|---|---|---|
| 2 | Обчислити, скільки днів треба копати город, якщо в середньому за день людина скопує 2 сотки. Кількість соток у городі вводить користувач. | Обчислити, скільки деталей робив робітник за годину, якщо він працював 8 годин. Кількість зроблених деталей вводить користувач. | Обчислити швидкість літака, якщо він летів від міста А до міста В 5 годин. Відстань між містами задає користувач. | Обчислити вартість комп'ютерів на складі, якщо один коштує близько 5000 гривень. Кількість комп'ютерів задає користувач. |
| 3 | Готуючись до нового навчального року учень купив С1 зошитів по 20 коп., С2 зошитів по 60 коп., С3 олівців по 15 коп., С4 ручок по 1 грн. 50 коп. Визначити, скільки було заплачено за кожний вид товару та загальну суму. | Чоловік вирішив вкрити стіни та підлогу ванної кімнати плиткою. Виміряв висоту стін А1, довжину стін А2 та А3. Розміри плитки 0,15 м на 0,15 м. Коштує 1 плитка 40 коп. Порахувати кількість плиток та їхню загальну вартість. | Людина зробила передплату преси на пошті. Було виписано газету за ціною 1,3 грн за номер (виходить А1 рази в місяць), журнал за ціною 3 грн за номер (виходить А2 рази в місяць) та газету за ціною 1,6 грн за номер (виходить А3 рази на місяць). Обчислити вартість передплати на рік. | Учень допомагав бабусі продавати фрукти на базарі. Він продав Х1 кг яблук за ціною 2 грн, Х2 кг груш за ціною 3,5 грн, Х3 кг слив за ціною 2,4 грн та Х4 кг абрикос за ціною 4 грн. Розрахувати, скільки грошей отримав учень за кожний вид фруктів та загальну суму. |

## Практична робота. Створення та реалізація лінійних програм

**Мета:** формування навичок складання, введення і редагування лінійних програм.

**Завдання:**

- *студенти повинні знати:* інтерфейс, призначення меню середовища програмування Turbo Pascal, структуру програми, синтаксис операторів введення-виведення, присвоєння;
- *студенти повинні вміти:* запускати середовище програмування, запускати програми на виконання, переглядати результат, зберігати програму, складати та виконувати в середовищі найпростіші лінійні програми.

**Обладнання:** дошка, зошит, середовище програмування.

**Хід виконання:**

Вивчити теоретичний матеріал з даної теми:

- запис констант, змінних, стандартних функцій;
- правила запису алгебраїчних виразів;
- використання вказівки присвоювання;
- організація введення та виведення даних.

**В-1**

1. Скласти програму обчислення значення заданої функції:

$$f = \frac{3\sin(a + \frac{b}{c})}{tg(c-b) + a^2} + bc$$

2. Виконати дану програму і обчислити значення функції при заданих даних:
   a = 3;
   b = 4,6;
   c = 7.
3. Скласти програму обчислення площі та периметра трикутника за трьома даними сторонами. Виконати тестування програми при a = 3, b = 4, c = 5.
4. Записати дві дані програми у звіт з практичної роботи.

**В-2**

1. Скласти програму обчислення значення заданої функції:

$$f = \frac{\sqrt{bc + a^2} + 5}{\ln(b)} + \sin(a)$$

2. Виконати дану програму і обчислити значення функції при заданих даних:
   a = 7,4;
   b = -6;
   c = 0.
3. З курсу фізики відомо, що сила взаємодії між двома зарядами прямо пропорційна добутку цих зарядів і обернено пропорційна квадрату відстані між ними, тобто

   $F = \frac{kq_1q_2}{r^2}$

   де k =9*109Нм2/Кл2. Обчислити силу, якщо:

$q_1$=0,03; $q_2$=0,25; r = 1,2

4. Записати дві дані програми у звіт з практичної роботи.

**В-3**

1. Скласти програму обчислення значення заданої функції:

$$f = \frac{-b + \sqrt{b^2 + 4ac}}{2a} + |c|$$

2. Виконати дану програму і обчислити значення функції при заданих даних:
   a = 6;
   b = 8;
   c = 3,5.
3. Радіус Місяця 1740 км. Складіть програму для обчислення площі поверхні $S = 4\pi r^2$ і об'єму планети $V = \frac{4}{3}\pi r^3$.
4. Записати дві дані програми у звіт з практичної роботи.

**В-4**

1. Скласти програму обчислення значення заданої функції:

$$f = ab + \frac{a^b - \cos(c)}{arctg(b)}$$

2. Виконати дану програму і обчислити значення функції при заданих даних:
   a = 0,1;
   b = 5,3;
   c = 7.
3. Задано координати вершин чотирикутника у площині: ($x_1$; $y_1$), ($x_2$; $y_2$), ($x_3$;$y_3$), ($x_4$; $y_4$). Складіть програму для обчислення довжини його діагоналей.
4. Записати дві дані програми у звіт з практичної роботи.

*Додаткове завдання:*

Уведіть тризначне ціле число. Складіть програму для обчислення добутку і суми його цифр.

**В-5**

1. Скласти програму обчислення значення заданої функції:

$$f = tg(c) - \sqrt{\frac{b}{c} + a}$$

2. Виконати дану програму і обчислити значення функції при заданих даних:
   a = 5;
   b = 2,4;
   c = 0,2.
3. Задано координати вершин чотирикутника у площині: ($x_1$; $y_1$), ($x_2$; $y_2$), ($x_3$;$y_3$), ($x_4$; $y_4$). Складіть програму для обчислення довжин його сторін.
4. Записати дві дані програми у звіт з практичної роботи.

*Додаткове завдання:*

Уведіть тризначне ціле число. Складіть програму для обчислення добутку і суми його цифр.

**В-6**

1. Скласти програму обчислення значення заданої функції:

$$f = \frac{b^c + 4ctg(a)}{\ln(b)} + bc$$

2. Виконати дану програму і обчислити значення функції при заданих даних:
   a = 5;
   b = -3;
   c = 4,5.
3. Скласти програму обчислення площі та периметра трикутника за трьома даними сторонами. Виконати тестування програми при a = 3, b = 4, c = 5.
4. Записати дві дані програми у звіт з практичної роботи.

*Додаткове завдання:*

Уведіть тризначне ціле число. Складіть програму для обчислення добутку і суми його цифр.

**В-7**

1. Скласти програму обчислення значення заданої функції:

$$f = \frac{b + \cos(c)}{3b + 4\sqrt{a + c}} - 6c$$

2. Виконати дану програму і обчислити значення функції при заданих даних:
   a = 4;
   b = -5,2;
   c = 1,1.
3. З курсу фізики відомо, що сила взаємодії між двома зарядами прямо пропорційна добутку цих зарядів і обернено пропорційна квадрату відстані між ними, тобто

   F = $\frac{kq_1q_2}{r^2}$

   де k =9*109Нм2/Кл2. Обчислити силу, якщо:
   $q_1$=0,03; $q_2$=0,25; r = 1,2
4. Записати дві дані програми у звіт з практичної роботи.

*Додаткове завдання:*

Уведіть тризначне ціле число. Складіть програму для обчислення добутку і суми його цифр.

**В-8**

1. Скласти програму обчислення значення заданої функції:

$$f = \frac{e^a + 2\ln(b + c)}{arctg(b)} + ab$$

2. Виконати дану програму і обчислити значення функції при заданих даних:
   a = -6;
   b = 6,4;
   c = 8,8.
3. Радіус Місяця 1740 км. Складіть програму для обчислення площі поверхні

$S = 4\pi r^2$ і об'єму планети $V = \frac{4}{3}\pi r^3$.

4. Записати дві дані програми у звіт з практичної роботи.

*Додаткове завдання:*

Уведіть тризначне ціле число. Складіть програму для обчислення добутку і суми його цифр.

**В-9**

1. Скласти програму обчислення значення заданої функції:

$$f = \frac{\sqrt{\frac{b}{a-b}}}{a^2+ab} - |b+4|$$

2. Виконати дану програму і обчислити значення функції при заданих даних:
   a = -2;
   b = -5,2;
   c = 6.
3. Задано координати вершин чотирикутника у площині: ($x_1$; $y_1$), ($x_2$; $y_2$), ($x_3$;$y_3$), ($x_4$; $y_4$). Складіть програму для обчислення довжини його діагоналей.
4. Записати дві дані програми у звіт з практичної роботи.

*Додаткове завдання:*

Уведіть тризначне ціле число. Складіть програму для обчислення добутку і суми його цифр.

**В-10**

1. Скласти програму обчислення значення заданої функції:

$$f = \frac{a-3tg(b)}{\ln(5c)+6} + b^a$$

2. Виконати дану програму і обчислити значення функції при заданих даних:
   a = 1,5;
   b = 8;
   c = 0,5.
3. Задано координати вершин чотирикутника у площині: ($x_1$; $y_1$), ($x_2$; $y_2$), ($x_3$;$y_3$), ($x_4$; $y_4$). Складіть програму для обчислення довжин його сторін.
4. Записати дві дані програми у звіт з практичної роботи.

*Додаткове завдання:*

1. Уведіть тризначне ціле число. Складіть програму для обчислення добутку і суми його цифр.

## Матеріали для організації самостійної позааудиторної роботи студентів

**Тема СРС:** ***Системи числення. Трансляція чисел з однієї системи числення в іншу***

**Кількість навчальних годин:** 2

**Мета, завдання самостійної позааудиторної роботи:** вироблення практичних навичок переведення чисел з однієї системи числення в іншу

**Завдання для самостійної роботи студентів:**

***Завдання 1.*** Опрацювати теоретичний матеріал з теми і дати відповіді на запитання:

1. Система числення: основні поняття.
2. Позиційна система числення.
3. Непозиційна система числення.
4. Правила переведення з десяткової системи числення в задану.
5. Правила переведення з заданої системи числення в десяткову.
6. Переведення з двійкової системи числення в десяткову через опосередкування в вісімковій системі числення.
7. Переведення з десяткової системи числення в двійкову через опосередкування в вісімковій системі числення.

***Завдання 2.*** Виконати завдання згідно власного варіанту:

***В-1***

Здійснити переведення:

1. $(363)_{10} \rightarrow (\ldots)_{рим}$;
   $(824)_{10} \rightarrow (\ldots)_{рим}$;
   $(513)_{10} \rightarrow (\ldots)_{рим}$.
2. $(1022)_3 \rightarrow (\ldots)_{10}$
   $(2A74,12)_{11} \rightarrow (\ldots)_{10}$
   $(2AB7,15)_{15} \rightarrow (\ldots)_{10}$
   $(245,56)_8 \rightarrow (\ldots)_{10}$
   $(110010,11)_2 \rightarrow (\ldots)_{10}$
   $(15,024)_6 \rightarrow (\ldots)_{10}$
3. Перевести задані числа в пятіркову, двійкову і дванадцяткову системи числення:
   - 29
   - 401
   - 0,38
   - 267,19
   - 492,45
4. Перевести $(11110010011011)_2$; $(10010100110010)_2$ в десяткову систему числення через опосередкування в вісімковій; шістнадцятковій системах числення.
5. Перевести $(2568)_{10}$; $(76903)_{10}$; $(3489)_{10}$ в двійкову систему числення через опосередкування в вісімковій; шістнадцятковій системах числення

***B-2***

***I.*** Здійснити переведення:

**1.** $(456)_{10} \rightarrow (\ldots)_{рим}$;
$(2113)_{10} \rightarrow (\ldots)_{рим}$;
$(809)_{10} \rightarrow (\ldots)_{рим}$.

**2.** $(1201)_{3} \rightarrow (\ldots)_{10}$
$(37A3,12)_{11} \rightarrow (\ldots)_{10}$
$(1B04,A1)_{15} \rightarrow (\ldots)_{10}$
$(413,24)_{8} \rightarrow (\ldots)_{10}$
$(101101,11)_{2} \rightarrow (\ldots)_{10}$
$(23,055)_{6} \rightarrow (\ldots)_{10}$

**3.** Перевести задані числа в пятіркову, двійкову і дванадцяткову системи числення:

- 31
- 503
- 0,76
- 354,82
- 439,45

**4.** Перевести $(10010011010011)_2$; $(10011001100100)_2$ в десяткову систему числення через опосередкування в вісімковій; шістнадцятковій системах числення.

**5.** Перевести $(4312)_{10}$; $(54309)_{10}$; $(3819)_{10}$ в двійкову систему числення через опосередкування в вісімковій; шістнадцятковій системах числення

***B-3***

Здійснити переведення:

**1.** .$(512)_{10} \rightarrow (\ldots)_{рим}$;
$(3015)_{10} \rightarrow (\ldots)_{рим}$;
$(916)_{10} \rightarrow (\ldots)_{рим}$.

**2.** $(1102)_{3} \rightarrow (\ldots)_{10}$
$(42A15,26)_{11} \rightarrow (\ldots)_{10}$
$(2B16,A4)_{15} \rightarrow (\ldots)_{10}$
$(627,23)_{8} \rightarrow (\ldots)_{10}$
$(101001,01)_{2} \rightarrow (\ldots)_{10}$
$(42, 035)_{6} \rightarrow (\ldots)_{10}$

**3.** Перевести задані числа в пятіркову, двійкову і дванадцяткову системи числення:

- 42
- 324
- 0,89
- 645,12
- 267,93

**4.** Перевести $(101001010010)_2$; $(110110110011)_2$ в десяткову систему числення через опосередкування в вісімковій; шістнадцятковій системах числення.

**5.** Перевести $(5243)_{10}$; $(8631)_{10}$; $(2469)_{10}$ в двійкову систему числення через опосередкування в вісімковій; шістнадцятковій системах числення

***В-4***

Здійснити переведення:

1. $(386)_{10} \rightarrow (\ldots)_{рим}$;
   $(2803)_{10} \rightarrow (\ldots)_{рим}$;
   $(472)_{10} \rightarrow (\ldots)_{рим}$.
2. $(2011)_3 \rightarrow (\ldots)_{10}$
   $(23A14,62)_{11} \rightarrow (\ldots)_{10}$
   $(12A5,B7)_{15} \rightarrow (\ldots)_{10}$
   $(574,73)_8 \rightarrow (\ldots)_{10}$
   $(110001,11)_2 \rightarrow (\ldots)_{10}$
   $(35,024)_6 \rightarrow (\ldots)_{10}$
3. Перевести задані числа в пятіркову, двійкову і дванадцяткову системи числення:
   - 25
   - 567
   - 0,98
   - 734,73
   - 315,29
4. Перевести $(111100111001)_2$; $(100011100100)_2$ в десяткову систему числення через опосередкування в вісімковій; шістнадцятковій системах числення.
5. Перевести $(4814)_{10}$; $(9764)_{10}$; $(5839)_{10}$ в двійкову систему числення через опосередкування в вісімковій; шістнадцятковій системах числення

**Тема СРС:** ***Системи числення. Трансляція чисел з однієї системи числення в іншу***

**Кількість навчальних годин:** 2

**Мета, завдання самостійної позааудиторної роботи:** вироблення практичних навичок виконання дій з двійковими числами.

**Завдання для самостійної роботи студентів:**

Виконати завдання:

а)111001+1001
б) 100111+111
в) 1111+10011
г) 1000+1101
д)10000-111
е) 11001-1001
ж) 110000-10001
з) 101*101
и) 1101*1100
і) 10011*111
ї) 1001:11
й)11001:101
к) 10001:111

**Тема СРС:** ***Елементи алгебри логіки***

**Кількість навчальних годин:** 2

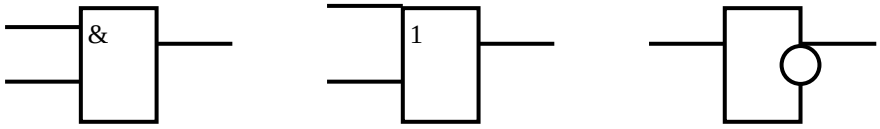
**Мета, завдання самостійної позааудиторної роботи:** ознайомлення з логічними елементами, вироблення навичок складання логічних схем.

**Теоретичні відомості**

Розглянемо, як і чому логічні операції використовуються в ЕОМ.

По-перше, логічні операції – найбільш прості, вони оперують однозначними двійковими цифрами, якими можна записати будь-яку інформацію. По-друге, з винайденням тріода (транзистора чи електронної лампи) виявилося, що логічні операції легко можна реалізувати у вигляді електронних схем.

Логічний елемент зображується на схемах у вигляді прямокутника з одним або двома входами і одним виходом. Елементи, відповідні вже відомим логічним операціям, позначаються на схемах так:



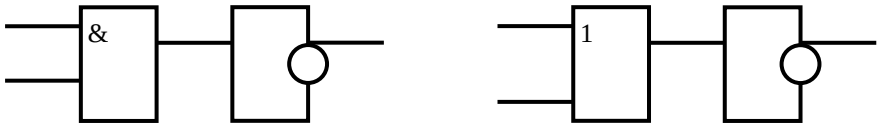

елемент ***І*** елемент ***АБО*** елемент ***НІ***

*(реалізує кон'юнкцію)* *(реалізує диз'юнкцію)* *(інвектор)*

В електроніці найбільш поширені елементи двох інших типів: ***І-НІ*** (*елемент Шеффера*) і ***АБО-НІ*** (*елемент Пірса*):


елемент ***І-НІ*** елемент ***АБО-НІ***

Ці елементи є "складними", тому що реалізуються через комбінацію елементів ***І***, ***АБО***, ***НІ***:




елемент ***І-НІ*** елемент ***АБО-НІ***

Випишемо таблиці істинності для елемента Шеффера (***І-НІ***) та елемента Пірса (***АБО-НІ***):

| **X** | **Y** | **X∧Y** | $\overline{\mathbf{X \wedge Y}}$ |
|---|---|---|---|
| 0 | 0 | 0 | ***1*** |
| 0 | 1 | 0 | ***1*** |
| 1 | 0 | 0 | ***1*** |
| 1 | 1 | 1 | ***0*** |

| **X** | **Y** | **X∨Y** | $\overline{\mathbf{X \vee Y}}$ |
|---|---|---|---|
| 0 | 0 | 0 | ***1*** |
| 0 | 1 | 1 | ***0*** |
| 1 | 0 | 1 | ***0*** |
| 1 | 1 | 1 | ***0*** |

**Базовий елемент** – це логічний елемент (наприклад *І-НІ*, *АБО-НІ*), який береться як основа при побудові складних схем, які є основою конструкції ЕОМ.

За допомогою елементів *І-НІ* і *АБО-НІ* можна реалізувати всі інші логічні операції ЕОМ. За допомогою основних базових елементів складаються вузли комп'ютера: шифратори, дешифратори, регістри, елементи пам'яті, суматори.

### *Теорія автоматів*

Розглянемо найпростіші (без елементів пам'яті) автомати – пристрої, які перетворюють вхідні потоки інформації, що складаються з одного чи багатьох сигналів, у вихідні.

Автомат можна розглядати як пристрій, який має певну кількість входів і виходів. Кожен з входів автомата може перебувати в одному із двох станів: 0 або 1. Кожен з виходів може перебувати в одному з двох станів: на виході є сигнал: 1, немає сигналу: 0.

Стан кожного із виходів певним чином залежить від комбінації сигналів, поданих на всі входи.

Автомат можна сконструювати, маючи логічну формулу, за допомогою елементарних пристроїв, які реалізують логічні операції *І*, *АБО*, *НІ*.

Розглянемо етапи створення автомата:

Задача $\Rightarrow$ логічна формула $\Rightarrow$ логічна схема $\Rightarrow$ фізична схема $\Rightarrow$ виготовлення пристрою

Розглянемо приклад.

Головний суддя має кнопку А, два інші – кнопки В і С. На табло з'явиться повідомлення "ШТРАФНИЙ УДАР" і пролунає свисток (це сигнал на виході автомата), якщо: рішення прийнято трьома суддями, або двома, серед яких є головний суддя (натискання суддями на кнопки – це вхідні сигнали). Скласти логічну формулу і функціональну (фізичну) схему автомата.

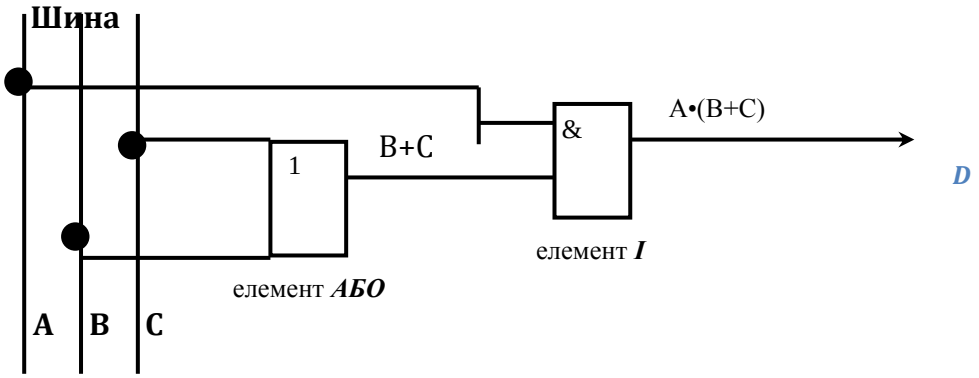
1. ***Задача*** перед нами поставлена.
2. Потрібно скласти ***логічну формулу***. Вхідні сигнали (аргументи) – А, В, С. Вихідний сигнал позначимо буквою D. Згідно умови складемо логічну формулу. Міркуємо: вирішальну роль відіграє головний суддя (А), але обов'язкова підтримка хоча б ще одного (В або С). Отже, А ***І*** (В ***АБО*** С) – така логічна формула забезпечує виконання умови задачі. Запишемо цю формулу зі звичними для нас знаками: D=A•(B+C). Перевіримо правильність цієї формули. За умовою D=1 в випадках, якщо: A=1, B=1, C=1; A=1, B=1, C=0; A=1, B=0, C=1.

   У всіх інших випадках D=0. Складемо таблицю істинності для перевірки:

| **A** | **B** | **C** | **B $\vee$ C** | **A $\wedge$ (B $\vee$ C)** |
|---|---|---|---|---|
| 0 | 0 | 0 | 0 | ***0*** |
| 0 | 0 | 1 | 1 | ***0*** |
| 0 | 1 | 0 | 1 | ***0*** |
| 0 | 1 | 1 | 1 | ***0*** |
| 1 | 0 | 0 | 0 | ***0*** |
| 1 | 0 | 1 | 1 | ***1*** |
| 1 | 1 | 0 | 1 | ***1*** |
| 1 | 1 | 1 | 1 | ***1*** |

   Отже, переконалися, що логічна формула складена правильно.

3. Наступний етап – складання ***логічної схеми***.




4. Маючи логічну схему можна скласти ***фізичну схему*** даного автомата. Для цього логічні елементи на фізичній схемі замінюються відповідними деталями (наприклад, діодами, транзисторами).
5. На цьому робота конструктора закінчується. Фізична схема відправляється на виробництво для серійного виготовлення автомата (пристрою).

**Мікроелектроніка** (інтегральна електроніка) – це область електроніки, яка пов'язана із створенням і застосуванням в ЕОМ вузлів і блоків, виконаних на інтегральних схемах і мікромініатюрних пристроях.

**Інтегральна схема** – це складна схема, яка містить тисячі чи навіть мільйони електронних елементів (діодів, транзисторів тощо).

**Завдання для СРС:**

1. ***Дати відповіді на запитання:***

1) Логічні величини та логічні операції: кон'юнкція, диз'юнкція, імплікація, заперечення.
2) Побудова таблиць істинності.
3) Еквівалентні висловлювання.
4) Основні базові логічні елементи.
5) Теорія автоматів.

2. ***Завдання:*** *Побудувати схему пристрою, який реалізує функцію:*
$\phi(x, y, z) = (X\overline{Y} \vee \overline{XYZ})\overline{Y}$

**Тема СРС:** ***Робота в середовищі програмування***

**Кількість навчальних годин:** 2

**Мета, завдання самостійної позааудиторної роботи:** ознайомлення з середовищем програмування Turbo Pascal.

**Завдання для самостійної роботи студентів:**

1. **Опрацювати теоретичний матеріал з теми і дати відповіді на запитання:**
   1) Поняття програми.
   2) Поняття про мови програмування. Класифікація мов програмування.
   3) Поняття про системи програмування.
   4) Поняття про інтерпретацію та компіляцію.
   5) Інтегровані середовища програмування. Поняття редактора, транслятора, налагоджувача.

6) Робота в середовищі програмування. Запуск програм на виконання.

**2. Виконати завдання:**

**1).** Вибрати правильну відповідь.

З допомогою команди **SAVE** можна:

а) Прочитати програму з файла на диску.
б) Записати програму в файл на диск.
в) Відредагувати програму.
г) Відкрити вікно для створення нового файла.

**2)**. Встановити відповідність між назвами команд меню системи програмування Pascal та результатами їх дії:

| | |
|---|---|
| 1) **Open** _____ | а) Записати програму в файл на диск |
| 2) **Save**_____ | б) Виконати програму |
| 3) **Save as** | в) Записати програму в файл на диск під новим іменем |
| 4) **Run**_____ | г) Відкрити вікно для створення нового файла |

**3).** Вказати порядок зчитування і редагування програми з файла на диску.

а) Внести необхідні доповнення до програми, виконати її і виправити помилки.
б) Записати програму в файл на диск, використавши команду **SAVE**.
в) Вибрати і активізувати команду **FILE.**
г) Вибрати і активізувати команду **OPEN.**
д) Ввійти в головне меню.
е) Натиснути клавішу **TAB**, вибрати з таблиці ім'я потрібного файлу і натиснути **ENTER**.

**4.** Обрати та записати правильну відповідь.

1) Щоб виконати програму, потрібно натиснути комбінацію клавіш _____
2) Щоб переглянути результати виконання програми, потрібно натиснути комбінацію клавіш _____
3) Щоб вийти з середовища програмування Turbo Pascal, потрібно натиснути комбінацію клавіш _____

а) Alt+X;
б) Ctrl+F9;
в) Alt+F5;
г) Alt+F9

**Тема СРС:** ***Типи даних і виразів в ТР, оператори мови Pascal***

**Кількість навчальних годин:** 2

**Мета, завдання самостійної позааудиторної роботи:** ознайомлення з алфавітом мови програмування; типами даних в Pascal, структурою програми, правилами запису виразів та використанням формул.

**Завдання для самостійної роботи студентів:**

**1.** Опрацювати теоретичний матеріал з теми і дати відповіді на запитання:

1) Мова програмування Паскаль: алфавіт, синтаксис.
2) Опис програм за правилами мови програмування.
3) Типи даних в Паскаль.
4) Запис виразів та використання функцій в Паскаль.
5) Поняття оператора: оператори введення-виведення, присвоєння.

**2.** Оберіть імена, які користувач утворив правильно:
а) a, a1, 2A, A+1, A99B, 55;
б) abc, ABC, A, **for**, **begin**;
в) i, ii, i-i, i_i, 2i, **var**;
г) Mynumber, My_number, My number, My- number, **do**;
д) aa, aa+aa, a:a, 4a, a4.

**3.** Запишіть імена, які можна утворити з трьох символів: a, b, 4.

**4.** Запишіть числа у звичайному вигляді:
6.25E+1, 3.5E-1, 25.5E+2, 0.85E-2?

**5.** Запишіть числа у форматі mEp:
0.0025, 0.00012, -0.000005.

**6.** Запишіть числа у звичайному вигляді:
0.1E+2, -35.5E-1, 4.0E-1, 55.0E+0?

**7.** Запишіть числа у форматі mEp, де m<1:
65, 125.5, 458.2, 250.

**8.** Оголосіть змінні N, i, number як цілі; A, suma, p — як дійсні.

**9.** Оголосіть змінні p, s як числові дійсні; Word1, kod — як текстові.

**10.** Якого значення набуде змінна A після обчислення виразів:
а) A := 5 + 2 * 3 - 1; (відповідь: 5 + 6 - 1 = 10);
б) A := (10 - 4) * 2 + 3;
в) A := 18 + 10 - 2 * 5;
г) A := 5 + 5 * 4;
д) A := 2*5 + 2*4 + 2* 3?

**11.** Які значення матимуть змінні після виконання команд присвоєння, якщо раніше виконувалися команди a: =4; b:=2:
а) a1 := a + 2 * b;
б) a2 := a * a + b * b;
в) a3 := a * b + 2;
г) a4:= a * (b - a)?

**12.** Що буде виведено на екран у результаті виконання програм:

**1)**
```
program one;
var a, b, c: integer;
begin
  a := 18;  b := 22;
  c := a + 2*b;
  write (c)
end.
```

**2)**
```
program two;
var s, v, t. : integer;
begin
  v := 60;  t := 4;
  s := v*t;
  write (s:4)
end.
```

**3)**
```
program three;
var a: integer;
begin
```

**4)**
```
program four;
var a, b, c: integer;
begin
```

```
  a := 3;
  a := a*a;
  a := a*a;
  write ('Це результат:', a:3)
end.
```

```
  a := (16+4)*2;
  b := 16+12*2;
  a:=a-b;
  write (a, a:5, b:6, 'Це результати')
end.
```

**5)**
```
program five;
var a, b: integer;
begin
  a := 1;  b := 1;
  a:= a+1; b:= b*a;
  write (a:2, b:4)
end.
```

**13.** Розгляньте приклад запису виразу в мові Паскаль та знаходження значення виразу в середовищі програмування.

Записати вираз

$$\frac{3\sin x + \cos 2x}{3{,}5 - 4|x|}$$

Відповідь: (3*sin(x) + cos(2*x))/(3.5 - 4 * abs(x)).

*Задача*. Функція y =f(x) задана виразом з прикладу. Обчислити значення цієї функції y деякій точці x, увівши значення x з клавіатури.

```
program Hometask;
uses Crt;
var x,y : real;
begin clrscr;
      write ('Введіть деяке число ');
      readln (x);
      y := (3*sin(x) + cos(2*x))/(3.5 - 4 * abs(x));
      writeln (x:6:2, y:6:2)
end.
```

**Тема СРС:** ***Оператори введення-виведення, присвоєння***

**Кількість навчальних годин:** 2

**Мета, завдання самостійної позааудиторної роботи:** ознайомлення з типами даних в Pascal, структурою програми, правилами запису виразів та використанням формул.

**Завдання для самостійної роботи студентів:**

1. Опрацювати теоретичний матеріал з теми і дати відповіді на запитання:
   - запис констант, змінних;
   - правила запису алгебраїчних виразів;
   - використання вказівки присвоєння;
   - організація введення і виведення даних.

- запис констант, змінних, стандартних функцій;
- правила запису алгебраїчних виразів;
- використання вказівки присвоювання;
- організація введення та виведення даних.

**2.** Складіть програми розв'язання задач згідно свого варіанту.

| Вар. № завд. | I | II | III | IV |
|---|---|---|---|---|
| 1 | Скласти програму розрахунку за формулою (*a, b, x* задає користувач) $y = \frac{2a + 4b}{x^2}$ | Скласти програму розрахунку за формулою (*a, b, x* задає користувач) $y = \frac{\sqrt{a} + \sqrt{x}}{\sqrt{b}}$ | Скласти програму розрахунку за формулою (*a, b, x* задає користувач) $y = \frac{b^3 - a^2}{\sqrt{x}}$ | Скласти програму розрахунку за формулою (*a, b, x* задає користувач) $y = \frac{\sqrt{x} + b^5}{\sqrt{a}}$ |
| 2 | Визначити, яку платню одержить робітник, якщо йому нараховано C грн., премія 10% та податок 13%. | Визначити суму грошей людини у банку через рік, якщо внесено B грн., приріст 12% в місяць. | Визначити масу поросятка масою M через 3 місяці, якщо за 1 місяць він набрав 5% ваги, за 2-й – 10%, за 3-й – 12%. | Вартість ремонту садиби K. Визначити кінцеву вартість, якщо протягом 2 місяців ремонту будматеріали дорожчали на 5% щомісячно. |

## Перелік питань для підготовки до Модульних контрольних робіт

### Модуль 1: Системи числення. Математичні основи роботи комп'ютера

1. Поняття системи числення.
2. Позиційна система числення.
3. Непозиційна система числення.
4. Правила переведення з десяткової системи числення в задану.
5. Правила переведення з заданої системи числення в десяткову.
6. Методи опосередкування.
7. Переведення з двійкової системи числення в десяткову через опосередкування в вісімковій системі числення.
8. Переведення з десяткової системи числення в двійкову через опосередкування в вісімковій системі числення.
9. Дії над двійковими числами.
10. Основні поняття математичної логіки.
11. Логічні операції: диз'юнкція, кон'юнкція, еквівалентність, заперечення, імплікація.
12. Формули алгебри висловлень.
13. Побудова таблиць істинності.
14. Поняття алгоритму.
15. Властивості алгоритмів.
16. Способи запису алгоритмів.
17. Базові структури алгоритмів.
18. Побудова блок-схем основних базових структур алгоритмів.
19. Основні логічні елементи.
20. Теорія автоматів.
21. Побудова логічних схем.

### Модуль 2: Основи мови програмування Паскаль та методологія програмування

1. Основні етапи розв'язування задачі з використанням ЕОМ.
2. Поняття мови програмування, програми.
3. Класифікація мов програмування.
4. Види програмування, їх характеристика.
5. Система програмування, її складові.
6. Середовище програмування Turbo Pascal: інтерфейс та основні прийоми роботи користувача.
7. Мова програмування Паскаль: алфавіт, синтаксис.
8. Опис програм за правилами мови програмування. Структура паскаль-програми.
9. Запис констант, змінних.
10. Типи даних в Паскаль.
11. Використання функцій при записі виразів в Паскаль.
12. Поняття оператора: оператори введення-виведення, присвоєння.
13. Оператор присвоєння, його синтаксис..
14. Оператори введення Read, Readln.
15. Оператори виведення Write, Writeln.

## Матеріали для організації індивідуальної роботи студентів

### Модуль 2. Основи мови програмування Паскаль та методологія програмування

**Кількість навчальних годин:** 6

**Мета, завдання індивідуальної позааудиторної роботи:** виконання роботи з опрацювання теми індивідуального дослідження, яка включає такі складові:

1) опрацювання літературних та інтернет-джерел для пошуку інформації згідно теми;
2) самостійне вивчення та опрацювання відібраного матеріалу;
3) оформлення роботи згідно вимог;
4) представлення та захист роботи.

**Тематика індивідуальних навчально-дослідницьких завдань:**

1. Форми представлення інформації в ЕОМ.
2. Машина Тюринга.
3. Комп'ютер і булева алгебра.
4. Форми подання інформації в комп'ютері.
5. Логічні елементи, що застосовуються в електронно-обчислювальній техніці.
6. Комбінаційні схеми. Побудова комбінаційних схем.
7. Інформаційні моделі. Комп'ютерне моделювання.
8. Теорія алгоритмів.
9. Алгоритми сортування.
10. Графічне представлення алгоритмів згідно з вимогами стандартів ЄСПД.
11. Ада Лавлейс – перша жінка-програміст.
12. Еволюція мов програмування.
13. Універсальні мови програмування.
14. Програмування: огляд основних понять.
15. Мова програмування Java Script.
16. Стандартні типи змінних. Опис типів змінних.
17. Лінійна програма на Турбо Паскаль. Структура, принцип виконання.
18. Поняття про оператори вводу-виводу.
19. Поняття про процедурний тип.
20. Поняття циклу. Оператори циклу.
21. Динамічні змінні і структури даних.
22. Структуровані типи даних в мові Паскаль і робота з ними.
23. Структурне програмування.
24. Принцип модульності, властивості модулів.
25. Місце мови Паскаль в сучасному програмуванні.

**Вимоги до представленої роботи:**

- орієнтований обсяг роботи: 7-10 сторінок;
- формат паперу - А4, всі поля – по 2 см;
- шрифт – TNR, розмір – 14 пт, міжрядковий інтервал – 1,5;
- на початку роботи – ЗМІСТ, обов'язковим є наявність висновків та списку використаних джерел;
- у ВСТУПІ необхідно вказати мету та завдання роботи;
- титульна сторінка оформляється згідно вимог;
- обов'язковою складовою ІНДЗ є наявність презентації з теми для використання під час захисту (5-10 слайдів).

## Рекомендована література


1. Архангельський А. Программирование в Delphi 6 / А. Архангельський. – М. : Бином, 2001.
2. Буров Є. Комп'ютерні мережі / Є. Буров. – Львів : БаК, 1999.– 468с.
3. Валецька Т. М. Комп'ютерні мережі. Апаратні засоби: Навч. посібник / Т. М. Валецька. – К : Центр навчальної літератури, 2004. – 208с.
4. Гаєвський О. Ю. Інформатика: 7-11 кл.: Навч. посіб. / О. Ю. Гаєвський. – К. : А.С.К., 2006.
5. Галіцин В.К. Багатокористувацькі обчислювальні системи і мережі: Навч. посібник / В.К.Галіцин, Ф.А.Левченко.– К : КНЕУ, 1998. – 360с.
6. Галкин В.А. Телекоммуникации и сети: Учебное пособие для вузов / В. А. Галкин, Ю. А. Григорьев.– М : МГТУ им. Баумана Н. Э., 2003.– 608с.
7. Глинський Я. М. Основи інформатики та обчислювальної техніки. – Частина II: “Комп’ютери” / Я. М. Глинський. – Львів : СП “БаК”, 1997.
8. Гуржій А. Н., Зарецька І. Т., Колодяжний Б. Г. Інформатика (підручник), 10-11 кл. / А. Н. Гуржій, І. Т. Зарецька, Б. Г. Колодяжний. – Х. : Факт, 2006.
9. Дарахвелидзе П. Г., Марков Е. П. Delphi – среда визуального программирования / П. Г. Дарахвелидзе, Е. П. Марков – С.-Пб. : BHV, 1996. – 352 с.
10. Жуков А. Изучаем Delphi / А. Жуков – С.-Пб. : Питер, 2002.
11. Забарна А, Войченко О. Візуальне програмування у DELPHI: Практикум / А. Забарна, О. Войченко — К. : Вид. дім «Шкіл. світ»: Вид. Л. Галіцина, 2006. –128 с.
12. Інформатика: Комп’ютерна техніка. Комп’ютерні технології : підручник для студ. вузів / В. А. Баженов, П. С. Венгерський, В. М. Горлач та ін. – [2-е вид.]. – К. : Каравела, 2011. – 591 с.
13. Караванова Т. П. Інформатика. Основи алгоритмізації та програмування (процедурне програмування): навч. посібник / Т. П. Караванова. – К. : Аспект, 2005.
14. Кулаков Ю. О., Луцький Г. М. Комп’ютерні мережі / Ю. О. Кулаков, Г. М. Луцький – К. : Юніор, 2003. – 395 с.
15. Культин Н. Б. Программирование в Turbo Pascal 7.0 и Delphi / Н. Б. Культин – С.-Пб. : BHV, 1998. – 240 с.
16. Левченко О. М., Завадський І. О., Прокопенко Н. С. Основи Інтернету. Навчальний посібник / О. М. Левченко, І. О. Завадський, Н. С. Прокопенко – К. : Видавнича група BHV, 2007. – 318 с.
17. Литвин І.І. Інформатика: теоретичні основи і практикум : підручник. – [2-ге вид., стереотип.] / І. І. Литвин, О. М. Конопчук, Ю. Д. Дещинський. – Львів : «Новий Світ – 2000», 2007. – 304 с.
18. Пєхота О. М. Освітні технології / О. М. Пєхота. – К. : А.С.К., 2001 – 256 с.

19. Ребрина В. А. Інформатика. Навчальний посібник, 10 кл. / В. А. Ребрина. – К. : Генеза, 2007.
20. Ривкінд Й. А., Лисенко Т. І. Інформатика 11 клас. Рівень стандарту: підручник / Й. А. Ривкінд, Т. І. Лисенко. – К. : Генеза, 2010.
21. Руденко В. Д., Макарчук О. М., Патланжоглу М. О. Курс інформатики (у 2-х ч.), (навчально-методичний посібник) / В. Д. Руденко, О. М. Макарчук, М. О. Патланжоглу. – К. : Фенікс, 2004.
22. Програмування в середовищі DELPHI. Методичні рекомендації до практичних робіт / [укл. Т. Г. Четверикова] – Луцьк: Терен, 2012. – 110 с.

***Інформаційні ресурси***

http: //www.mon.gov.ua
http://www.osvita.info/
http://informatic.org.ua/
http://itosvita.ucoz.ua/
http://osvitaonline.googlepages.com/
http://galanet.at.ua/
http://kafinfo.org.ua/
http://www.spohelp.ru/
http://www.osvita.ht.ua/
http://www.metod-kopilka.ru/
http://kpolyakov.narod.ru/
http://kafinfo.org.ua/korysni-posylannya
http://videouroki.net/
http://fo.ru/
http://www.ucoz.ru/
http://www.kan-studio.ru/
http://ki.at.ua/
http://blog.gvmir.com/

## Для нотаток

**УПРАВЛІННЯ ОСВІТИ І НАУКИ**
**ВОЛИНСЬКОЇ ОБЛАСНОЇ ДЕРЖАВНОЇ АДМІНІСТРАЦІЇ**
**ЛУЦЬКИЙ ПЕДАГОГІЧНИЙ КОЛЕДЖ**

***Циклова комісія викладачів фізико-математичних дисциплін***

**Навчально-методичне забезпечення**

**дисципліни «Програмування з методикою навчання» (І семестр)**

***Навчально-методичний посібник***

Підписано до друку 15.02.2015
Ум. друк. арк. 9,75. Зам № 584. Тираж 100.
Папір офсетний. Гарнітура Cambria. Друк офсетний.
Друк ПП Іванюк В. П. 43021, м. Луцьк, вул. Винниченка,65.
Свідоцтво Держкомінформу України
ВЛн № 31 від 04.02.2004 р.

**Навчально-методичне забезпечення дисципліни «Програмування з методикою навчання» (І семестр) :** Навчально-методичний посібник / [Укл. Т. Г. Четверикова] – Луцьк, 2015. – 72 с.

Навчально-методичний посібник містить методичні матеріали з дисципліни «Програмування з методикою навчання»: тексти лекцій, розробки практичних занять, матеріали для організації самостійної позааудиторної та індивідуальної роботи студентів, перелік питань для підготовки до модульних контрольних робіт.

Матеріали, подані у посібнику, стануть у нагоді студентам педагогічних коледжів, які здобувають освіту за спеціальністю 5.01010201 «Початкова освіта» з додатковою кваліфікацією «Вчитель інформатики в початковій школі».

**УДК 004.41**
**ББК 32.973.2-018**
**О-75**