

УДЗ

УСІ ДОМАШНІ ЗАВДАННЯ

8
клас

Розв'язання всіх завдань
до всіх підручників

АЛГЕБРА

НІМЕЦЬКА МОВА

ГЕОГРАФІЯ

ФІЗИКА

ІНФОРМАТИКА

Видавництво



™ Підручники
і посібники >

Частина 1

**Розв'язання
усіх вправ і завдань
до підручника
«ІНФОРМАТИКА.
8 клас»
Морзе Н. В. та ін.**



РОЗДІЛ 1. КОДУВАННЯ ДАНИХ

1. Кодування даних

Обговорюємо

1. Кодування повідомлень — процес перетворення за певними правилами одного подання набору даних в інший. Під час кодування повідомлення відбувається зміна вигляду повідомлення без зміни його змісту.
3. Двійковим називають кодування повідомлень з використанням двох сигналів. Набір даних, отриманий у результаті двійкового кодування, називають двійковим кодом.
5. Цифри 0 і 1 у двійковому коді повідомлення мають назву біт. Послідовність із восьми бітів називають байтом.
6. Для кодування одного символу використовують двійковий код, довжина якого дорівнює 1 байту, тобто 1 символ = 1 байт = 8 бітів.
10. Довжина двійкового коду повідомлення — це кількість бітів чи байтів у двійковому коді цього повідомлення.
14. Таблиця кодів символів ASCII містить коди літер лише англійського алфавіту. Для кодування літер інших алфавітів було розроблено інші таблиці, наприклад, KOI8-U (KOI — код обміну інформацією) і Windows-1251 містять без змін усі коди таблиці ASCII, а також коди літер кирилиці. Таблиця Юнікод складається з 17 наборів по 65536 значень кодів, що дає можливість закодувати майже всі символи писемності всіх світових мов.

Працюємо в парах

1. Запустити табличний процесор **MS Excel 2010** (Пуск ⇒ Всі програми ⇒ **Microsoft Office 2010** ⇒ **Microsoft Office Excel 2010**) ⇒ увести в діапазон клітинок A1:F2 текст за зразком ⇒ у клітинку A2 ввести кількість бітів ⇒ у клітинку B2 ввести формулу =A2/8, C2 — =B2/1024, D2 — =C2/1024, E2 — =D2/1024, F2 — =E2/1024. Інша таблиця переведення: ввести у клітинку A1 текст кілобайт, B1 — біт, C1 — байт, D1 — мегабайт, E1 — гігабайт, F1 — терабайт ⇒ у клітинку A2 ввести кількість кілобайтів ⇒ у клітинку B2 ввести формулу = A2*1024*8, C2 — =B2/8, D2 — =A2/1024, E2 — =A2/1024, F2 — =A2/1024.

2.

	П	р	и	в	і	т	,					
Unicode	U+041F	U+0440	U+0438	U+0432	U+0456	U+0442	U+0020	U+0020				
Windows-1251	207	240	232	226	179	242	130	160				
	д	р	у	ж	є	!		Ї	и			
Unicode	U+0434	U+0440	U+0443	U+0436	U+0435	U+0021	U+0020	U+0427	U+0438	U+0020		
Windows-1251	228	240	243	230	229	33	160	215	232	160		

	п	о	д	о	б	а	с	т	ь	с	я	
Unicode	U+031F	U+043E	U+0434	U+043E	U+0431	U+0430	U+0454	U+0442	U+044C	U+0441	U+044F	U+0020
Windows-1251	239	238	228	238	225	224	186	242	252	241	255	160
	т	о	б	і								
Unicode	U+0442	U+043E	U+0431	U+0456	U+0020	U+	U+	U+	U+	U+	U+	U+
Windows-1251	242	238	225	179	160							
	і	н	ф	о	р	м	а	т	и	к	а	?
Unicode	U+0456	U+043D	U+0444	U+043E	U+0440	U+043C	U+0430	U+0442	U+0438	U+043A	U+0430	U+003F
Windows-1251	179	237	244	238	240	236	224	242	232	234	224	136

Довжина двійкового коду повідомлення, яке закодоване за допомогою таблиці Unicode, дорівнює 752 біт або 94 байти, а довжина двійкового коду повідомлення, яке закодоване за допомогою таблиці Windows-1251, дорівнює 376 бітів або 47 байтів.

Працюємо самостійно

1. $24 \cdot 16 = 384$ біти або $384 : 8 = 48$ байтів.
2. 8192 біти = 1024 байти = 1 Кб.
3. 2097152 байти : $1024 = 2048$ Кб : $1024 = 2$ Мб.
4. Ні, не достатньо, відсутні дані про систему кодування.
5. Оскільки перекодування кожного символу повідомлення з Unicode у KOI-8 зменшилося на 8 бітів, то в текстовому повідомленні було $480 : 8 = 60$ символів.
6. Обчислимо кількість символів на одній сторінці: $128 \cdot 48 = 6144$ (символи). Оскільки довжина двійкового коду книжки становить 720 Кб, то це становить $720 \cdot 1024 = 737\,280$ (байтів). Кожен символ кодувався 2-байтовим кодуванням Unicode, тому було затрачено $737\,280 \cdot 16 = 11\,796\,480$ байт, а всього у книжці було $11796480 : 6144 = 1920$ (сторінок).

РОЗДІЛ 2. АПАРАТНО-ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ КОМП'ЮТЕРА

3. Апаратне забезпечення комп'ютера

Обговорюємо

1. Етапи розвитку засобів обчислень: давнє приладдя для лічби (зарубки, вузлики, лічильні палички); немеханічні обчислювальні пристрої (абак, логарифмічна лінійка); механічні обчислювальні пристрої (механічний арифмометр); електронні обчислювальні машини (комп'ютер).

2. Під поколінням ЕОМ розуміють усі типи й моделі електронних обчислювальних машин, розроблені різними користувацькими колективами, але побудовані за одними й тими ж науковими і технічними принципами.
3. Архітектура ПК — принципи роботи та взаємодії основних пристроїв комп'ютера: процесора, внутрішньої та зовнішньої пам'яті, а також пристроїв введення і виведення даних.
4. Основні складові комп'ютера фон Неймана: пристрій управління, арифметико-логічний пристрій, пам'ять, пристрої введення та виведення даних.
5. Процесор — це мікросхема, яка створюється на напівпровідниковому кристалі (або кількох кристалах) шляхом застосування складної мікроелектронної технології. Складові мікропроцесора: арифметико-логічний пристрій для виконання арифметичних і логічних операцій з даними; пристрій управління для забезпечення функціонування всіх складових комп'ютера; реєстри власної пам'яті.
6. Пам'ять комп'ютера поділяють на внутрішню та зовнішню. Основною властивістю пам'яті є її ємність. Ємність пам'яті вимірюється в байтах, кілобайтах, мегабайтах, гігабайтах, терабайтах тощо.
7. До пристроїв зовнішньої пам'яті належать накопичувачі на жорстких магнітних дисках (НЖМД), пристрої для роботи з оптичними дисками, пристрої, що використовують флеш-пам'ять. Сучасні НЖМД мають такі властивості: ємність; діаметр жорстких дисків; швидкість обертання (6 Тбайт–2000 Гбайт; 3,5–2,5 дюйма; 5400–7200 об/хв та 5400 об/хв відповідно для настільних комп'ютерів та для ноутбуків). Пристрої для роботи з оптичними дисками характеризуються ємністю (640 Мбайт–100 Гбайт) та можливістю зчитування, запису і перезапису даних (CD-, DVD-, BD-ROM; CD-, DVD-, BD-R; CD-, DVD-RW, BD-RE). Пристрої флеш-пам'яті, SSD-дисків мають велику ємність і високу швидкість запису та зчитування даних.
8. Внутрішня пам'ять служить для забезпечення роботи процесора, пристрої внутрішньої пам'яті розміщуються на материнській платі. Пристрої зовнішньої пам'яті кріпляться до неї за допомогою різних з'єднань, на відміну від пристроїв внутрішньої пам'яті (за винятком постійної пам'яті) призначені для довготривалого зберігання даних, після вимкнення живлення дані з носіїв зовнішньої пам'яті не зникають.
9. Пристрої введення та виведення даних призначені для введення даних до комп'ютера та виведення результатів їх опрацювання у вигляді, зручному для користувача.
10. До складу мультимедійного обладнання входять звукова карта, акустична система, накопичувачі для оптичних дисків, мікрофон, відеокамера, відеопрогравач, цифрова фотокамера.

Працюємо в парах

1. (все в купі, від 1-6 питання) До пристроїв введення даних належать клавіатура, сканер, фотокамера, графічний планшет, мікрофон, відеокамера, веб-камера, ТВ-тюнер, миша, тачпед, мультимедійна дошка, сенсорний екран, джойстик, геймпед, кейпед, тощо.
До пристроїв виведення даних належать монітор, принтер, плотер, навушники, звукові колонки, мультимедійні проектори.

Клавіатури класифікують за призначенням: на стандартні (для настільних ПК), компактні, мультимедійні, ігрові тощо; за типом підключення до системного блоку: дротові або бездротові (радіо, Blue-tooth, Wi-Fi).

Маніпулятори миші у комп'ютерному класі здебільшого мають такі властивості: за призначенням — для настільних ПК; за типом підключення — дротові; за типом датчиків руху — оптичні; за кількістю кнопок — 2 або 3; за типом корпусу — симетрична стандартна.

Сканери, що використовуються в комп'ютерних класах школи, найчастіше за принципом дії — з протяжним механізмом. Основними властивостями таких сканерів є розмір області сканування, роздільна здатність (від 600 до 6400 точок на дюйм), швидкість сканування — 5–20 секунд.

Монітор є основним пристроєм для виведення даних у персональних комп'ютерах. Для опрацювання графічних даних, що виводяться на екран монітора, у комп'ютерах використовується спеціальний пристрій — *відеоадаптер*. Відеоадаптер містить спеціальний графічний процесор та додаткову оперативну пам'ять — графічну (відео) пам'ять. Розрізняють графічні адаптери за типом процесора (наприклад, Radeon R7 370, GeForce GTX 950), обсягом пам'яті (від 1 до 6 Гбайт і більше), системою під'єднання до материнської плати та монітора тощо.

Основні властивості *принтера* для домашнього друку: розмір матеріалу — формат А4, кількість кольорів — монохромний або багатокольоровий, роздільну здатність — не менше 1200 точок на дюйм, швидкість друку — від 1 до 50 сторінок за хвилину, тип підключення до комп'ютера — дротове або бездротове.

7. Властивостей процесора.

<i>Властивості</i>	<i>Значення властивостей процесора</i>		
Модель	Intel Core i7-5960 Extreme Edition	AMD FX-8350	AMD Athlon X4 86
Тактова частота	3000 МГц	4000 МГц	3700 МГц
Кількість ядер	8	8	4
Розрядність	64	64	64

Працюємо самостійно

- Етапи розвитку інформаційних технологій: 1 — етап ручних засобів (до середини XV ст.), 2 — етап механічних засобів (середина XV ст. – XIX ст.), 3 — етап електричних засобів (середина XIX ст. – до 40-х років XX ст.), 4 — етап електронних засобів (від 40-х років XX ст. до наших днів). На кожному етапі розвитку інформаційних технологій засоби опрацювання повідомлень змінювались таким чином: 1 етап — мозок і пальці людини, ручні обчислювальні прилади (абак, рахівниця), 2 етап — друкарські машини, фотоапарати, арифмометри, машинки для обчислень, 3 етап — фонографи, грамофони, електричні друкарські машинки, табулятори, електричні арифмометри, магнітофони, ротапринти, 4 етап — електронні калькулятори, комп'ютер.

5. Карта знань «Пристрої комп'ютера».

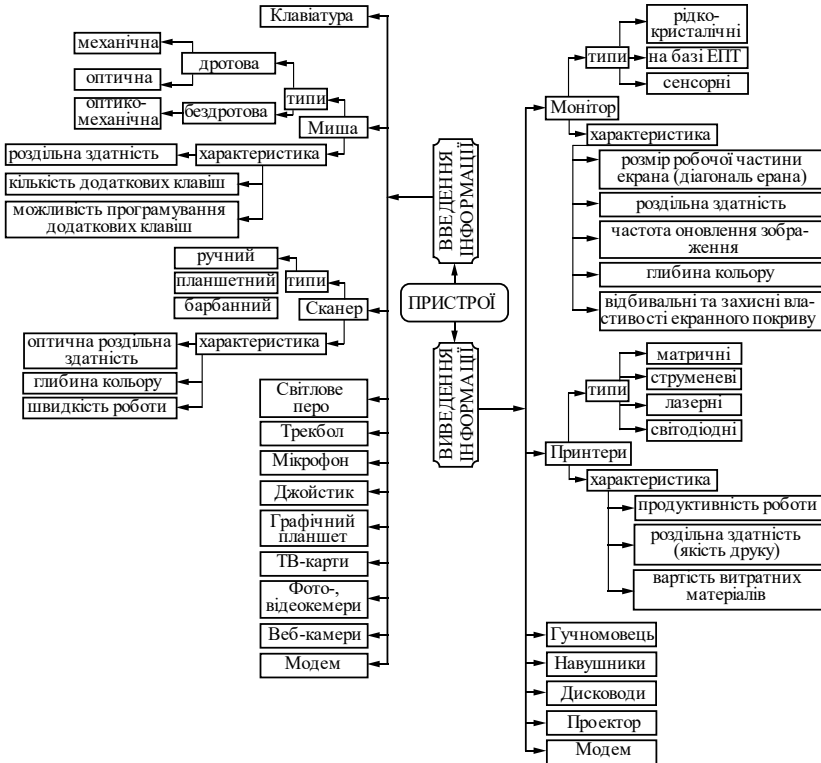


Рис. 3.1. Карта знань

6. Найкращим варіантом домашнього ПК, який можна використовувати як мультимедійний центр, буде моноблок з великою діагоналлю монітора, обов'язково наявність багатьох роз'ємів (для підключення мікрофону, проектора, акустичної системи тощо) або портативний ПК (ноутбук) з аналогічними характеристиками. Для прикладу розглянемо технічні характеристики моноблоку Asus ET2702IGTH-B017N.

Процесор	Чотирьох ядерний Intel Core, i5-4460S (2,9 ГГц)
Дисплей	27 WQHD (2560×1440)
Об'єм оперативної пам'яті	8 ГБ
Тип пам'яті	DDR3 1600 МГц
Модуль WiFi	Наявний
Тип відеокарти і об'єм відеопам'яті	Дискретна, AMD Radeon HD 8890A, 2 ГБ виділеної відеопам'яті

Чіпсет материнської плати	Intel B85
Об'єм HDD	2 ТБ
Оптичний привід	DVD-Super-Multi
Порти	Роз'єми на бічній панелі: 4×USB 3.0; 1×вхід для мікрофона, вихід на навушники, лінійний вихід, кардрідер 3-в-1. Роз'єми на задній панелі: 2×USB 2.0, 1 HDMI - вхід, HDMI – вихід, LAN (RJ45, роз'єм сабвуфера (2,5 мм)). Роз'єм для замка Kensington, роз'єм живлення.
Бездротові технології	Bluetooth 4.0, Wi-Fi 802.11 b/g/n/ac
Аудіо	Sonic Master Premium
Вихідна потужність	4×3 Вт
Встановлене ПЗ	Windows 8
Маса	13,5 кг
Розміри	660×508×233 мм
Додаткові можливості	Веб-камера 2.0 Мп, вбудований мікрофон
Комплект поставки	Asus Eee Top, клавіатура, миша, адаптер, живлення, кабель живлення, болти для кріплення на стіну

5. Програмне забезпечення комп'ютера

Обговорюємо

1. Під програмним забезпеченням розуміють сукупність програм і правил, а також документації щодо функціонування комп'ютера для опрацювання даних.
3. Програми базового рівня зберігаються у спеціальних мікросхемах постійного запам'ятовуючого пристрою та утворюють базову систему введення-виведення — BIOS.
5. Програми системного рівня забезпечують взаємодію інших програм комп'ютера з програмами базового рівня та безпосередньо з апаратним забезпеченням.
4. Програми службового рівня взаємодіють як з програмами базового рівня, так і з програмами системного рівня.
5. Програмне забезпечення прикладного рівня — комплекс прикладних програм, за допомогою яких виконуються конкретні завдання.
7. Основні види ліцензій: вільні, відкриті, власницькі.
8. Операційна система — це програмний комплекс, що забезпечує такі функції: управління ресурсами — злагоджену роботу всіх апаратних засобів комп'ютера; управління процесами — виконання всіх програм та їх взаємодію з пристроями комп'ютера та даними; взаємодію між користувачем та комп'ютером.
9. Ліцензія на програмне забезпечення — правовий документ, що визначає правила використання та поширення програмного забезпечення.

9. Схема класифікації операційних систем.

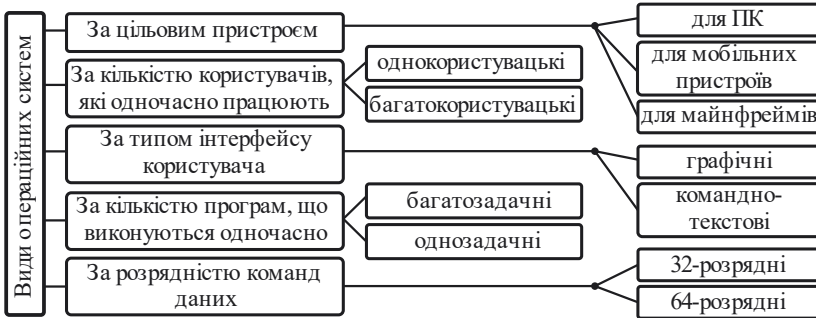


Рис. 5.1. Види операційних систем

- Архівування даних виконується для об'єднання кількох файлів (каталогів) в один файл — архів. Програми-архіватори не лише компонують файли в єдиний архів, а також використовують алгоритмічне перекодування інформації для зменшення загального об'єму пам'яті, який вони займають.
- Стиснення даних застосовують для більш раціональної передачі та зберігання даних на пристроях шляхом алгоритмічного перетворення даних з метою зменшення займаного об'єму. Стиснення з частковою втратою даних застосовують у тих випадках, коли таке стиснення (зменшення об'єму) є пріоритетним, а повна відповідність вихідних і відновлених даних не потрібна. Частіше таке стиснення використовують для зменшення обсягу аудіо- та відеоданих, цифрових фотографій тощо.

Працюємо в парах

- Піраміду з указаних в умові програм можна скласти так: 1) операційна система; 2) драйвер модема; 3) прикладні програми забезпечення доступу до Інтернету; 4) антивірусні програми; 5) текстовий редактор.
- Правило розпакування архіву: двічі клацнути ліву клавішу мишки (ЛКМ) на архіві ⇒ у вікні програми-архіватора натиснути кнопку **Видобути** ⇒ відкриється вікно **Шляхи і параметри видобування**, вказати місце, куди архів буде розпаковано (якщо не вказати шлях розпакування, то файли з архіву будуть поміщені в теж місце, де знаходиться і сам архів) ⇒ натиснути кнопку **ОК**.
- Існує кілька розповсюджених причин, чому не на всіх комп'ютерах можна пограти у певні ігри: 1) ПК не відповідає мінімальним системним вимогам, зазначеним в описі гри (гра або взагалі не запуститься або буде зависати чи видавати різноманітні помилки); 2) ПК заражено вірусами (програми працюють не коректно, ігри зокрема, і вся система загалом); 3) не вистачає програмного забезпечення для запуску конкретної гри (у таких випадках потрібно доставити певне програмне забезпечення); 4) не встановлені драйвери пристроїв.

Працюємо самостійно

- Рейтинг операційних систем на початок 2016 року: I місце — Windows 7; II місце — Windows XP; III місце — Windows 8.1; IV місце — Windows 10; V місце — Windows Vista.

сце — OS X 10.11 (продовження MacOS); VI місце — Windows 8; VII місце — OS X 10.10; VIII місце — Linux. Рейтинг мобільних операційних систем за 2016 рік: I місце — Android, II місце — iOS. Світова статистика відповідає загальній ситуації в Україні. Рейтинг використання Windows XP залишається значним, оскільки вона встановлена у 98% банкоматів у США (в Україні теж на більшості банкоматів встановлена дана операційна система). На моєму домашньому ПК була встановлена Windows 7, яку ми оновили до Windows 10, на моєму телефоні та планшеті встановлена ОС Android (на більшість гаджетів, а тепер вже і на нові покоління флагманів Samsung встановлено саме ОС Android).

Для побудови діаграми в табличному процесорі знадобляться числові дані (подані нижче). Тип діаграми для відображення рейтингу операційних систем за даними 2016 року доречно подати у вигляді стовпчастої, секторної або гістограми.

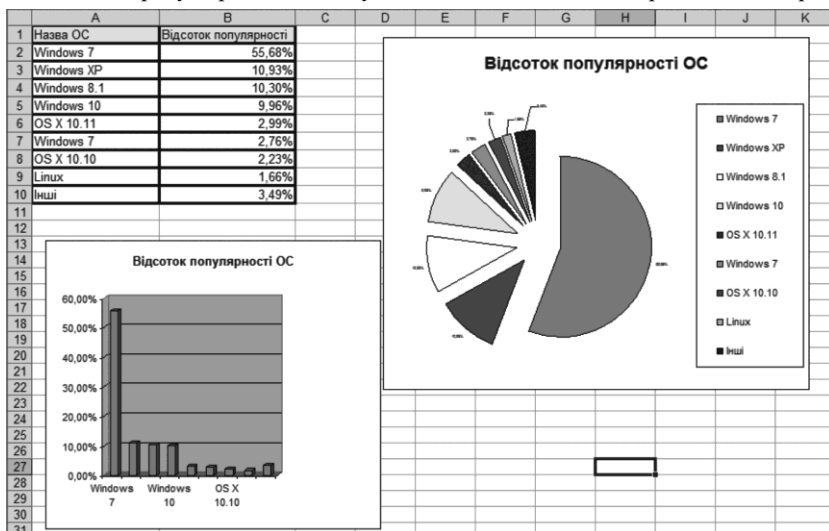


Рис. 5.2. Приклади діаграм

4. Таблиці для кодування за алгоритмом Хаффмана слів «молоко», «кукуріку».

<i>Таблиця ймовірності символів</i>		<i>Розрахунок коду Хаффмана</i>
Символ	Значення (частота входження)	Значення
о	3	0
м	1	011
л	1	010
к	1	00

<i>Таблиця ймовірності символів</i>		<i>Розрахунок коду Хаффмана</i>
Символ	Значення (частота входження)	Значення
к	3	11
у	3	0
р	1	101
і	1	100

Довжина коду та коефіцієнт стиснення даних за алгоритмом Хаффмана: в результаті кодування файлу, де символ А зустрічається 10 разів, В — 20, С — 30, D — 5, Е — 25, F — 10.

У результаті кодування отримуємо: С = 11 (2 біти), А = 0111 (4 біти), Е = 10 (2 біти), F = 011 (3 біти), В = 00 (2біти), D = 0110 (4 біти).

Кожен символ спочатку представлявся 8 бітами (1 байт), а оскільки в результаті кодування було зменшено число біт, необхідних для представлення кожного символу, відповідно тепер зменшено і розмір вихідного файлу. Схема стиснення відображене у таблиці:

<i>Частота входження</i>	<i>Початкові значення (біт)</i>	<i>Отримання значення за алгоритмом Хаффмана</i>	<i>Зменшено на (біт)</i>
С — 30	$30 \cdot 8 = 240$	$30 \cdot 2 = 60$	180
Е — 25	$25 \cdot 8 = 200$	$25 \cdot 2 = 50$	150
В — 20	$20 \cdot 8 = 160$	$20 \cdot 8 = 40$	120
А — 10	$10 \cdot 8 = 80$	$10 \cdot 4 = 40$	40
F — 10	$10 \cdot 8 = 80$	$10 \cdot 4 = 40$	50
D — 5	$5 \cdot 8 = 40$	$5 \cdot 4 = 20$	20
Усього (біт)	800	240	560

Початковий розмір файлу: 100 байт — 800 біт;

Розмір стиснутого файлу: 30 байт — 240 біт;

Результат стиснення файлу — зменшення обсягу на 70% ($560 : 800 \cdot 100\% = 70\%$).

РОЗДІЛ 3. ОПРАЦЮВАННЯ ТЕКСТОВИХ ДАНИХ

7. Текстовий документ і його об'єкти

Обговорюємо

- У текстовому документі Word 2007 можна створити списки таких типів: маркований — кожний елемент списку на початку першого рядка абзацу починається певним символом (маркером); нумерований — на початку першого рядка абзацу вказується його номер; багаторівневий — елементи списку можуть вкладатись у інші списки.
- Щоб розмістити текст у кілька колонок, треба виділити потрібний текст, виконати Розмітка сторінок \Rightarrow Параметри сторінки \Rightarrow Колонки, встановити у вікні Колонки кількість колонок. Для колонок можна змінити такі властивості: ширину, відстань між колонками, наявність розділювача. Ширину колонок, відступи між колонками, кількість колонок можна встановити так: виконати вказівку Розмітка сторінок \Rightarrow Параметри сторінки \Rightarrow Колонки \Rightarrow Інші колонки, встановити відпо-

відні параметри, після встановлення значень властивостей колонок натиснути ОК.

5. Для того, щоб вставити у текст символи, які відсутні на клавіатурі (математичні, латинські, графічні), використовують список таких символів, який відкривається вибором кнопки **Символ** на вкладці **Вставлення** в групі **Символи**. Їх редагування та форматування здійснюється за правилами роботи з усіма символами у MS Word 2007.
 7. До операцій форматування графічних об'єктів належать: змінення форми і розмірів об'єкта, кольору, типу та шаблону ліній його контуру, кольору й ефектів заливки. Для форматування графічних примітивів зручно користуватися елементами керування тимчасової вкладки **Формат**, яка з'являється на Стрічці в розділі **Засоби малювання** під час виділення графічного об'єкта.
 8. Способи створення таблиць в текстовому документі: 1) вставити таблицю простої структури; 2) накреслити таблицю довільної структури; 3) перетворити фрагмент тексту на таблицю; 4) вставити таблицю з колекції шаблонів. Таблицю складної структури можна накреслити в документі, для цього виконати такі вказівки: 1) виконати **Вставлення** ⇒ **Таблиці** ⇒ **Таблиця** ⇒ **Накреслити таблицю**; 2) намалювати олівцем контури всієї таблиці; 3) намалювати лінії, що розділяють рядки та стовпці.
- 13-14.** Прихованими називають символи, які вводяться користувачем з клавіатури, але не виводяться на папір під час друкування, і зазвичай, не відображаються на екрані. Але якщо включити режим відображення цих символів, то у відповідних місцях тексту з'являться спеціальні позначення цих символів. Приклади прихованих символів: пропуск, кінець абзацу, табуляція. Їх використовують, щоб краще розуміти структуру тексту чи документа.

Працюємо в парах

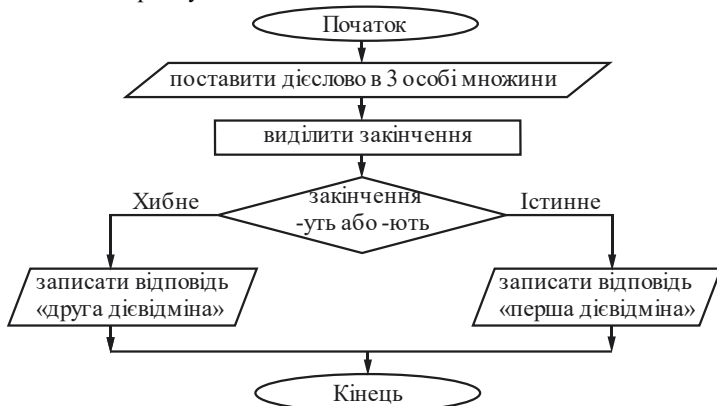
1. Порівнюючи значки файлів, збережених в різних форматах (Рис. 7.1) можна зробити висновок, що для позначення формату того чи іншого текстового файлу використовуються різні значки. Ця різниця зроблена для того, щоб користувачі візуально ідентифікували формати текстових файлів та розуміли, в якій програмі його відкрити і які операції над даними в файлі можна виконати.
2. Нумерований і маркований список використовується для позначення переліку. Спільними параметрами при зміні нумерованого та маркованого списку для користувачів є можливість скористатись бібліотекою маркерів / бібліотекою нумерованих списків та змінити рівень списку. Відмінність в доступних налаштуваннях маркованого списку полягає в тому, що можна визначити новий маркер (з символів, рисунків), а для нумерованого списку є опція визначення нового числового формату та можливість встановити / змінити значення нумерації.
3. Інколи виникає необхідність використання колонок при оформленні документів у текстовому процесорі. Наприклад, порівняння показників до якогось значення, створення газетоподібних документів, оформлення бланків тощо.
4. Недруковані символи призначені виключно для форматування, щоб користувач міг побачити місця вставлення пробілів (позначається крапкою), знаків табуляції (позначаються стрілками), перехід на новий рядок (позначається як знак абзацу), розрив сторінки, розрив розділу. Відповідно недруковані символи можна прихо-

вати і вони ніяк не впливають на зміст документа. А спеціальні символи доступні на вкладці **Вставка** у групі **Символи** кнопка **Символ**.

- Гра «Формула»: корені квадратного рівняння \Rightarrow сила тяжіння \Rightarrow швидкість \Rightarrow закон Ома \Rightarrow електричний опір провідника \Rightarrow щільність речовини \Rightarrow формула Герона.
- Переміщувати таблицю в інше місце документа можна з використанням маркера виділення таблиці (чотирьох направлена стрілка у лівому верхньому куті). Можна переміщувати частини таблиці в інше місце, виділивши необхідні рядки, вирізати їх, вставити курсором у необхідне місце в документі та вставити (цю ж саму операцію можна виконати за допомогою гарячих клавіш **CTRL+X**, **CTRL+V**)
- Вміст окремої клітинки можна відформатувати окремо, змінивши заливку, межі, вирівнювання в клітинці; змінивши текст, його кольори та напрям.
- Доцільність використання того чи іншого способу створення таблиці в текстовому процесорі: якщо треба перетворити в таблицю готовий текст, то використовують вказівку **Перетворити на таблицю**; якщо треба виконати обчислення у таблиці, то зручніше використати електронну таблицю Excel — доступ на вкладці Вставка група таблиці; якщо у таблиці треба оформити календар, зручніше скористатись експрес-таблицями; з допомогою вказівки **Вставка таблиці** / **Вставити таблицю** можна обрати або задати кількість рядків, стовпців, які є необхідними для побудови таблиці; коли треба побудувати таблицю складної структури, то використовують вказівку **Накреслити таблицю**.

Працюємо самостійно

- Блок-схема алгоритму визначення дієвідміни дієслова:



9. Опрацювання складних текстових документів

Обговорюємо

- Стиль — це набір значень властивостей певного типу об'єктів, який має власне ім'я. Їх використовують для завдання структурованого вигляду тексту, оформлення документів за певними правилами, виділення різного смислового забарв-

лення тексту з допомогою елементів форматування, з урахуванням основних вимог дизайну.

2. Зміст документа — перелік назв структурних частин документа із зазначенням номерів відповідних сторінок. Зміст використовують для швидкого переміщення по документу, пошуку потрібних структурних частин — розділів, пунктів.
3. Зазвичай у текстовому документі є тільки один розділ і всі сторінки документа відформатовано однаково (розмір аркуша, полів, орієнтація сторінки). Однак під час роботи з багатосторінковими документами може виникнути потреба, щоб деякі сторінки документа мали інші параметри. Наприклад, змінити орієнтацію окремої сторінки чи розмістити певний фрагмент у колонки. Таке оформлення можливе під час розподілу документа на розділи. Створення розділів бажано виконувати в режимі перегляду документа Розмітка сторінки і при включеному режимі відображення прихованих символів. Далі потрібно виконати Розмітка сторінки ⇒ Параметри сторінки ⇒ Розриви і вибрати у списку потрібний тип у переліку Розриви розділів. У місці розриву розділів з'являється недрукований символ Розрив розділу.
4. Існує кілька видів розривів розділів:
 - з наступної сторінки — розділ починається з наступної сторінки;
 - з поточної сторінки — розділ починається поточному місці документа;
 - з парної сторінки — розділ починається з наступної парної сторінки;
 - з непарної сторінки — розділ починається з наступної непарної сторінки.
5. Колонтитули використовуються для відображення в документах додаткової службової інформації, що уточнює або доповнює дані змісту основного тексту. У колонтитули можна вставити номери сторінок, назву документа або поточного розділу, прізвище автора, графічні зображення.
11. Структура документа — це ієрархічна схема розміщення складових частин документа. Найчастіше структуру створюють для багатосторінкових документів, розділяючи їх на розділи, глави, параграфи, пункти. Приклади: реферати, курсові роботи, брошури, книжки тощо.
Шаблони — відформатовані певним чином документи заготовки, які зберігаються в окремих файлах і використовуються як зразок для створення нових документів. Шаблон використовують для створення нових документів за певним зразком, наприклад, календар, грамота, подяка,звіт.
- 13?. Шаблон може встановлювати такі властивості документа: для сторінок — орієнтацію аркуша, розмір аркуша, поля; для абзацу — інтервали, вирівнювання, відступи; для символу — шрифт основного тексту, розмір, колір, інтервал. Існують такі типи шаблонів: *інстальовані* — шаблони певних типів документів, які інстальовані у складі пакету Microsoft Office 2007; *Microsoft Office Online* — шаблони різних типів документів, які розташовані на веб-сайті Microsoft Office Online; *шаблони користувача* — шаблони, які створені користувачем.
12. Шаблон відрізняється від документа способом використання: шаблон — це заготовка документа з готовими елементами тексту та оформлення, яка призначена для подальшого заповнення даними, а документ — це вже підготовлений текст.
13. Шаблони зберігаються у файлах з розширенням **dotx** або **dotm**. Приклади шаблонів: квитанції, звіти, календарі, формуляри, довідки, грамоти тощо.

Працюємо в парах

2. Внести зміни до будь-якого шаблону можна у такий спосіб: відкрити шаблон / створити на його основі новий документ ⇒ внести відповідні зміни ⇒ зберегти новий документ в потрібному форматі. Набір шаблонів учня: розклад уроків на тиждень, графік відвідування секцій, факультативів та гуртків, щоденник з поза-класного читання, шаблон оформлення реферату, вітальна листівка «З Днем народження».
3. Для створення власного шаблону виконати: створити документ ⇒ обрати команду Зберегти як вкладки Файл ⇒ вказати місце зберігання, ім'я файлу зі списку Тип файлу обрати Шаблон Word (якщо не вказати місце зберігання, то Office автоматично перейде до папки Налаштовані шаблони / або Користувацькі шаблони Office папки Документи). Користувацький шаблон у діалоговому вікні відображається в категорії Особисті.
4. Колонтитули використовуються для відображення в документах додаткової службової інформації, що уточнює або доповнює дані змісту основного тексту. У колонтитули можна вставити номери сторінок, назву документа або поточного розділу, прізвище автора, графічні зображення. Для створення колонтитулів використовуються елементи керування вкладки **Вставлення** групи **Колонтитули**. Відображаються колонтитули в документі тільки в режимах **Розмітка сторінки** та **Читання**. Розрізняють декілька видів оформлення колонтитулів для різних сторінок документа: колонтитули на першій сторінці, який буде відрізнятися від колонтитулів на інших сторінках; перша сторінка без колонтитулів; різні колонтитули на парних і непарних сторінках; різні колонтитули для розділу або параграфа.
5. Зміст документа — перелік назв структурних частин документа із зазначенням номерів відповідних сторінок. Зміст використовують для швидкого переміщення по документу, пошуку потрібних структурних частин — розділів, пунктів. Структура документа — це ієрархічна схема розміщення складових частин документа. Найчастіше структуру створюють для багатосторінкових документів, розділяючи їх на розділи, глави, параграфи, пункти. Приклади: реферати, курсові роботи, брошури, книжки тощо.

РОЗДІЛ 4. ОПРАЦЮВАННЯ ОБ'ЄКТІВ МУЛЬТИМЕДІА

11. Опрацювання аудіо- та відео файлів

Обговорюємо

1. Аудіофайли можуть зберігатися у файлах різних форматів (типів): WAV, WAVE, MIDI, FLAC, MP3. Файли з розширенням wav, mid, flac не передбачають втрати даних.
- 4?. Відеофайли можуть зберігатися у файлах різних форматів з розширенням mp4, wmv, mov, flv.
3. Програми для опрацювання мультимедійних даних поділяють на програми для перегляду або прослуховування (плеєри), програми для запису (захоплення) звуку і відео (грабери, рекордери), програми для конвертації даних (конвертори), музичні та відео редактори, музичні та відео студії.

1. Сервіси для розміщення мультимедійних файлів в Інтернеті: YouTube, Яндекс.Відео, Dailymotion.com, BBC.
8. Приклади музичних редакторів та студій — Audacity, Free Audio Editor, Wave Editor та Adobe Audition, Linux MultiMedia Studio, Psychedelics Psyche, Steinberg Cubase, Cakewalk Sonar.
6. Приклади відео редакторів та студій — Кіностудія Windows, Open Shot, Virtual Dub та Pinnacle Studio, Kino, Adobe Premiere Pro, Avid Media Composer.
7. Кіностудія Windows призначена для створення відеофільмів, що можуть містити фрагменти відео, фотографії, звуковий супровід (дикторський текст), титри.

Працюємо в парах

1. У Вікіпедії відомості про аудіоформати та відеоформати подані українською, російською та англійською мовами, вони є змістовно аналогічними. Українською мовою подано найбільш стисло відомості про ці поняття. Аудіо файл (шукати як «цифрові звукові формати / цифровий аудіоформат») — формат подання звукових даних, який використовується для цифрового звукозапису, подальшого зберігання записаного матеріалу на комп'ютері та інших електронних носіях інформації (аудіо файл — комп'ютерний файл, який містить звукозапис, складається з інформації про амплітуду і частоту звуку). Відео формат (шукати «відео» або «список форматів файлів» ⇒ «відео») — формат для запису, копіювання, збереження, відтворення аудіовізуальної інформації.
2. Переваги прослуховування аудіо записів онлайн: доступність; широкий вибір музичних стилів; можливість завантаження музики на ПК; зручна пошукова система за виконавцями або назвою твору. Недоліки прослуховування аудіо записів онлайн: необхідна наявність високошвидкісного Інтернету; не на всіх музичних порталах гарна якість аудіозаписів.
3. Моя улюблена радіостанція «Хіт FM» та «KISS FM», на сайті доступна можливість прослуховування ефіру онлайн.
4. Імпортувати відеозаписи у форматі .flv до проекту, що створюється в Кіностудії не можна. Необхідно спочатку конвертувати його в інший відео формат (WMV, AVI, M4V, MPEG).
5. Для ознайомлення можна скористатись наступною програмою-конвертером Freemake Video Converter. Вона має можливість імпортувати більшість популярних відео / аудіо / графічних форматів та переводить їх в AVI, MP4, WMV, SWF, 3GP, DVD, BLU-RAY, MP3. Також програма Freemake Video Converter здійснює відео підтримку для різних мультимедійних пристроїв (iPod, iPhone, iPad, Xbox, Sony PlayStation, Blackberry, Android та інших).

Працюємо самостійно

1. Назви аудіо-, відеоформатів: MPEG (англ. Moving Picture Experts Group — укр. Експертна група з питань рухомого зображення); AA (англ. Audible Audio Book File — укр. Звуковий Файл Аудіо Книги (в основному використовується для запису аудіо книг, які продаються через Інтернет-магазини Audible.com, iTunes Store)); WAVE, WAV (англ. Wavefarm Audio File Format — укр. У формі хвилі); SWF (англ. Small Web Format — укр. Невеликий Веб-формат, використовується для флеш-анімації, векторної графіки, відео й аудіо в мережі Інтернет); 3GP (англ. Third Generation Partnership Project (3GPP) — укр. Проект Партнерства

Третього Покоління, формат аудіо та відео); MKV — відео (з субтитрами і звуком); MKS — для субтитрів і MK3D — для 3D-відео (англ. Matroska — Матрьошка, помилковий варіант Матроска).

2. Ознайомлення з матеріалами <http://www.elecard.com/mpeg/faq/index.php>:
1) «Група MPEG збирається приблизно чотири рази на рік десь на тиждень»;
2) «Потік кодованого зображення рідко перевищує величину закодованого потоку в MPEG більше, ніж у 30 разів»;
3) «Кольорове цифрове зображення з стиснутої послідовності переводиться в колірний простір YUV (YCбCr). Компонент Y являє собою інтенсивність, а U і V — кольоровість. Оскільки людське око менш сприйнятливим до кольоровості, ніж до інтенсивності, то дозвіл колірних компонент може бути зменшено в 2 рази по вертикалі або і по вертикалі, і по горизонталі. До анімації і високоякісного студійного відео зменшення дозволу не застосовується для збереження якості, а для побутового застосування, де потоки більш низькі, і апаратура дешевша, така дія не призводить до помітних втрат у візуальному сприйнятті...».
3. Вся послідовність створення та розміщення подкасту в Інтернеті описана в категорії «Як створювати подкаст?» на веб-ресурсі <http://podfm.ru/topodcasters>.
4. Приклади відеозаписів на YouTube, що відображають зміст професій:
<https://www.youtube.com/watch?v=CgFHquzqDsA> — ландшафтний дизайнер;
<https://www.youtube.com/watch?v=U6CxoMhy5Xg> — архітектор;
https://www.youtube.com/watch?v=aukezY_-zdk — кондитер;
<https://www.youtube.com/watch?v=UpOkIrhuvCg> — повар;
<https://www.youtube.com/watch?v=Ywwu2dyEZ5w> — програміст.

12. Створення та налаштування відео й аудіо

Обговорюємо

1. Канал для розміщення відеофайлів на сайт YouTube створюють за таким алгоритмом: 1) Відкрити у браузері головну сторінку сервісу. 2) Вибрати у лівій частині вікна посилання Мій канал. 3) Увести у вікні, що відкрилося, логін і пароль пошти Gmail. 4) Увести ім'я та прізвище, під яким ваші відеоматеріали будуть розміщуватися в YouTube (за замовчуванням будуть використані дані з екаунта Gmail). 5) Ознайомитися з умовами використання сервісу за відповідним посиланням. 6) Вибрати кнопку Створити канал. 7) Вибрати кнопку Завантажити у верхній правій частині вікна. 8) Вибрати у списку Конфіденційність один з варіантів доступу до перегляду файлу: Для всіх, Не для всіх або Приватне. 9) Вибрати кнопку Завантажити і потрібний файл для розміщення. 10) Дочекатися завершення завантаження та ввести значення властивостей вашого відео у вікні налаштувань: назву відео, опис відео, ключові слова (теги), адреси друзів або кіл спілкування. 11) Вибрати кнопку Готово.
2. Під час завантаження відео на YouTube користувач задає значення наступних властивостей: конфіденційність, категорія, ліцензія, назва, опис, теги, коментарі, рейтинги, формати оголошень, мова відео, причина відсутності титрів, пропозиції покращити відео, місце зйомки, статистика відео.
3. Кіностудія Windows призначена для створення відеофільмів, що можуть містити фрагменти відео, фотографії, звуковий супровід (дикторський текст), титри.

4. До основних об'єктів вікна програми Кіностудія Windows відносять: 1) Панель швидкого запуску; 2) Стрічка; 3) Вікно перегляду; 4) Область монтажу; 5) Повзунок змінення масштабу області монтажу; 6) Відеодоріжка; 7) Звукова доріжка; 8) Кнопка керування переглядом; 9) Повзунок перегляду.
5. Область монтажу в програмі Кіностудія Windows призначена для відображення порядку монтажу кадрів фільму, що створюється засобами студії, позначення меж кадрів, відеоряду та звукового ряду, коментарів диктора.
6. Створення відеофільму складається з таких етапів: 1) створення сценарію — детальний план послідовності окремих кадрів фільму і розміщення окремих об'єктів; 2) підготовка матеріалів — зйомка всіх епізодів відповідно до сценарію, добір музики, фото, малюнків, запис звукового супроводу; 3) монтаж — опрацювання та структурування окремих кадрів для отримання цілісного відеофільму(у процесі монтажу матеріал редагується, уточнюється сценарій, накладається звук і титри).
7. Сценарій відеофільму — це детальний план послідовності окремих об'єктів. Він створюється для деталізації і спрощення процесу зйомки відео та подальшого монтажу.
8. Для того, щоб вставити до проекту відеофільму відео або фото з файлу, треба використати вкладку Головна, елементи керування групи Додавання, та вибрати необхідний об'єкт.
9. Для запису дикторського тексту треба використати елемент Записати дикторський текст групи Додавання вкладки Головна; звук дикторських коментарів відображається в окремий звуковий ряд.
10. Сутність узгодження відео та звуку полягає у співставленні по часовій шкалі звукових та відеорядів у процесі монтажу фільму. У Кіностудія Windows узгодження виконується у Області монтажу.

Працюємо в парах

1. Способи обтинання відео в програмі Кіностудія: 1) розділення відео — вкладка Редагування ⇒ група Редагування ⇒ команда **Розділити** ⇒ клацнути мишкою на місці встановлення розділення і Вікні проекту (після розділення відео можна поміняти місцями його частини, змінити порядок їх відтворення тощо); 2) засіб обтинання — вкладка Редагування ⇒ група Редагування ⇒ команда Засіб обтинання ⇒ контекстна вкладка Обтинання ⇒ команда Установити точку початку ⇒ перемістити маркер обтинання в необхідне місце, під Вікном попереднього перегляду, задавши точку кінця ⇒ переглянути отриманий результат, натиснувши кнопку **Відтворити**, у Вікні попереднього перегляду ⇒ натиснути кнопку **Зберегти обтинання**.

Працюємо самостійно

2. Відео, створене в програмі Кіностудія, можна розмістити на наступних сервісах: OneDrive, Facebook, YouTube, Vimeo, Flickr.

РОЗДІЛ 5. ОСНОВИ ПОДІЙНО ТА ОБ'ЄКТНО-ОРІЄНТОВАНОГО ПРОГРАМУВАННЯ

15. Мова та середовище програмування

Обговорюємо

1. Мова програмування — це система позначень для точного опису алгоритму, який потрібно виконати за допомогою комп'ютера.
2. Мова програмування має такі складові:
 - 1) Алфавіт — множина символів, з яких можна утворювати слова і речення цієї мови.
 - 2) Словник — набір слів, які використовуються в цій мові.
 - 3) Синтаксис — правила складання і запису мовних конструкцій: не словникових слів і речень.
 - 4) Семантика — встановлене однозначне тлумачення мовних конструкцій, правил їх виконання.
4. Транслятор — програма, яка записує команди програми, описаної деякою мовою програмування, машинною мовою.
4. Середовище Lazarus складається з таких складових: текстового редактора — для введення і редагування тексту програм; компілятора — для перекладу програми з мови програмування Object Pascal на мову команд, які може виконати процесор комп'ютера; засобів налагодження програми — для пошуку помилок в ній; довідкової системи.
5. Серед основних її переваг Python можна назвати такі: чистий синтаксис (для виділення блоків слід використовувати відступи); переносність програм (що властиве більшості інтерпретованих мов); стандартний дистрибутив має велику кількість корисних модулів (включно з модулем для розробки графічного інтерфейсу); можливість використання Python в діалоговому режимі (дуже корисне для експериментування та розв'язання простих задач); стандартний дистрибутив має просте, але разом із тим досить потужне середовище розробки, яке зветься IDLE і яке написано на мові Python; зручний для розв'язання математичних проблем (має засоби роботи з комплексними числами, може оперувати з цілими числами довільної величини, у діалоговому режимі може використовуватися як потужний калькулятор); відкритий код (можливість редагувати його іншими користувачами).

Працюємо в парах

1. Відкрити вікно браузера ⇒ ввести у поле адреси <https://blockly.demo/appspot.com/statac/code/index.html> ⇒ у верхньому правому куті обрати українську мову ⇒ спосіб 1 — у меню Блоки обрати Петлі ⇒ у переліку вибрати блок для виконання певних дій декілька разів ⇒ у відповідне поле для введення кількості повторів ввести 3 ⇒ вставити команду, яку потрібно виконати, для цього ⇒ у меню Блоки обрати Текст ⇒ у переліку вибрати блок для друкування заданого тексту, числа або інших значень ⇒ у відповідне поле ввести текст (наприклад, Привіт! Як справи?) ⇒ натиснути у верхньому правому куті кнопку для запуску програми на виконання ⇒ перевірити виконання про-

грами; спосіб 2 — три рази вставити блок для друкування заданого тексту, числа або інших значень ⇒ у відповідне поле ввести текст ⇒ натиснути у верхньому правому куті кнопку для запуску програми на виконання ⇒ перевірити виконання програми. У Скретч можна використовувати такі блоки: говорити <текст>, повторювати <3>.

- Відкрити вікно браузерера ⇒ ввести у поле запити «Історія виникнення мов програмування» ⇒ прочитати інформацію ⇒ створити лінію часу (зручно для створення використовувати графічні редактори, редактори карт знань, текстовий та табличний процесори тощо).
- IDLE — підсвічення синтаксису, вбудована система відлагодження, область виконання має декілька режимів. CodingCround — безкоштовна, не треба встановлювати на ПК, великий перелік мов програмування. PyCharm — декілька режимів області виконання, доступний список службових команд з підказками, візуалізація процесу тощо.

Працюємо самостійно

1.

	<i>Скретч</i>	<i>Спільне</i>	<i>Lazarus</i>	
Відмінне	Створення програми шляхом побудови блоків, мова програмування Logo	Розташування об'єктів на сцені	Створення програми шляхом написання коду, мова програмування Object Pascal	Відмінне
		Виконання подій		
		Область сценаріїв		
		Візуальні компоненти		
		Розділення вікон		

2. 1) print (125/25+10) відповідь: 15; 2) print (2*3*4*5/6) відповідь: 20; 3) print (765/9+48*4-121) відповідь: 156.

4. Завантажити середовище PyCharm (Пуск ⇒ Всі програми ⇒ JetBrains ⇒ JetBrains PyCharm 2016.1.4, який знаходиться на Робочому столі) ⇒ у меню File обрати команду New Project ⇒ у діалоговому вікні Create Project, що відкрилося обрати папку збереження проекту та задати ім'я проекту «Термінал» ⇒ натиснути кнопку Create ⇒ у меню File обрати команду New ⇒ у списку New File, що відкрився вибрати Python File ⇒ натиснути кнопку ОК ⇒ у вікні редактора коду ввести текст програми:

```
Name1=input('Введіть назву оператора мобільного зв'язку')
```

```
Name2=input('Введіть номер телефону для поповнення рахунку')
```

```
Name3=input('Введіть суму поповнення рахунку')
```

```
Print ('Шановний абоненте', name1, '! Ви поповнюєте телефон за номером', name2, 'на суму', name3, 'грн')
```

У меню Run обрати команду Run ⇒ перевірити правильність виконання програми.

16.Об'єкти програм із графічним інтерфейсом

Обговорюємо

1. Створювати об'єкти графічного інтерфейсу в середовищах програмування можна двома способами: 1) безпосередньо в редакторі коду програми;

- 2) використовуючи засоби графічного інтерфейсу середовища візуального програмування.
2. Проект — це набір файлів, з якими користувач працює під час створення прикладної програми в об'єктно-орієнтованому середовищі.
 3. Розробка прикладної програми в середовищі Lazarus складається з декількох етапів: 1) аналіз та планування виконання завдання; 2) підготовка проекту; 3) розміщення компонентів інтерфейсу користувача на формі проекту; 4) написання програми опрацювання подій; 5) тестування та налагодження програми.

Працюємо в парах

1. project1.exe — відкомпільований проект; project1.res — файл ресурсів проекту; project1.lpr — файл з початковим кодом програми; project1.lpi — файл проекту Lazarus; unit1.pas — вихідний файл модуля форми; unit1.lfm — дані форми. Деякі файли мають однакове ім'я, але різні розширення, оскільки є файли, які відповідають за опис форм, за програмні модулі та за головний файл проекту.
2. Навчальне середовище програмування Скретч підтримує подійно-орієнтоване програмування, тому що воно є динамічним, об'єкту можна присвоювати певну команду для виконання, об'єкти мають властивості.

Працюємо самостійно

1. Створення програми у середовищі PyCharm: завантажити середовище PyCharm (Пуск ⇒ Всі програми ⇒ JetBrains ⇒ JetBrains PyCharm 2016.1.4, який знаходиться на Робочому столі) ⇒ у меню File обрати команду New Project ⇒ у діалоговому вікні Create Project, що відкрилося обрати папку збереження проекту та задати ім'я проекту «Нагадування» ⇒ натиснути кнопку Create ⇒ у меню File обрати команду New ⇒ у списку New File, що відкрився вибрати Python File ⇒ натиснути кнопку OK ⇒ у вікні редактора коду ввести текст програми:

```
import tkinter
main= tkinter.Tk ()
label= tkinter.Label (text = «Необхідно дотримуватися часу роботи за ПК»)
label.pack()
main.mainloop()
```

У меню Run обрати команду Run ⇒ перевірити правильність виконання програми.

Створення програми у середовищі Lazarus: завантажити Lazarus (Пуск ⇒ Всі програми ⇒ Lazarus) ⇒ у вкладці Проект обрати Створити проект ⇒ підвести вказівник миші до зображення напису (TLabel) на панелі інструментів на вкладці Standard ⇒ два рази натиснути ліву клавішу миші ⇒ у властивостях напису в полі Caption змінити текст за замовчуванням на «Необхідно дотримуватися часу роботи за ПК» ⇒ на панелі інструментів натиснути кнопку Запустити (або F9) ⇒ перевірити правильність виконання програми ⇒ зберегти проект (у вкладці Файл обрати команду Зберегти як ⇒ обрати місце збереження проекту ⇒ ввести ім'я проекту ⇒ натиснути Зберегти).

Відповідь на два запитання: де більше часу витрачено і яка форма цікавіша?

2. Завантажити Lazarus (Пуск ⇒ Всі програми ⇒ Lazarus) ⇒ у вкладці Проект обрати Створити проект ⇒ підвести вказівник миші до зображення напису (TLabel) на панелі інструментів на вкладці Standard ⇒ два рази натиснути ліву клавішу миші ⇒ у властивостях напису в полі Caption змінити текст за замовчуванням на «Сьогодні о 18.00 наша родина іде на концерт» ⇒ на панелі інструментів натиснути кнопку Запустити (або F9) ⇒ перевірити правильність виконання програми ⇒ зберегти проект (у вкладці Файл обрати команду Зберегти як ⇒ обрати місце збереження проекту ⇒ ввести ім'я проекту ⇒ натиснути Зберегти).
3. Запустити Lazarus (Пуск ⇒ Всі програми ⇒ Lazarus) ⇒ у вкладці Проект обрати Створити проект ⇒ підвести вказівник миші до зображення напису (TLabel) на панелі інструментів на вкладці Standard ⇒ два рази натиснути ліву клавішу миші (аналогічно зробити ще два написи — Label2, Label3) ⇒ у властивостях напису Label1 в полі Caption змінити текст за замовчуванням на «Проект — набір файлів, з якими користувач працює під час створення прикладної програми», Label2 — «Екранна форма — вікно, яке користувач розробляє засобами середовища чи мови програмування», Label3 — «Інтерфейс користувача — спосіб взаємодії користувача з комп'ютером» ⇒ на панелі інструментів натиснути кнопку Запустити (або F9) ⇒ перевірити правильність виконання програми ⇒ зберегти проект (у вкладці Файл обрати команду Зберегти як ⇒ обрати місце збереження проекту ⇒ ввести ім'я проекту ⇒ натиснути Зберегти).
4. Запустити Lazarus (Пуск ⇒ Всі програми ⇒ Lazarus) ⇒ у вкладці Проект обрати Створити проект ⇒ підвести вказівник миші до зображення напису (TLabel) на панелі інструментів на вкладці Standard ⇒ два рази натиснути ліву клавішу миші ⇒ у властивостях напису Label1 в полі Caption змінити текст за замовчуванням на «Листопад 2016» ⇒ у властивостях в полі Font натиснути та обрати Шрифт, Накреслення, Розмір, Ефекти, Колір тощо ⇒ аналогічним чином створити Label2 та використовуючи комбінації клавіш Ctrl+C — копіювати, Ctrl+V — вставити, створити підписи днів, числа ⇒ на панелі інструментів натиснути кнопку Запустити (або F9) ⇒ перевірити правильність виконання програми ⇒ зберегти проект (у вкладці Файл обрати команду Зберегти як ⇒ обрати місце збереження проекту ⇒ ввести ім'я проекту ⇒ натиснути Зберегти).

18. Властивості та методи екранної форми і елементів управління

Обговорюємо

1. Для зміни значень властивостей можна використати динамічний спосіб. Коли зміна відбувається у процесі виконання програми за допомогою команди присвоєння.
2. Щоб описати дії, які мають виконуватися при настанні події для об'єкта, необхідно написати програмний код, який в середовищі Lazarus розпочинається службовим словом `procedure`.
6. Кнопки, які створюються у середовищі Lazarus за допомогою компонента `Button`, використовують, щоб під час виконання програми після їх натиснення виконувалася деякий набір команд.

7. Метод — фрагмент програмного коду, який вбудований в об'єкт та передбачає виконання деяких дій з ними.

Працюємо в парах

1. Переваги: зміна значення властивостей до кожного об'єкта, список доступних властивостей та дії з ними під час введення. Недоліки: опис конкретної події заноситься у відповідний фрагмент, потрібно уважно писати ім'я об'єкту, до якого застосовується зміна значення властивостей.
2. Яким може бути ім'я другого об'єкта (1), як називається перший об'єкт (Label — напис), як називаються 2, 3, 4-ий об'єкти (Button — кнопка), як змінити значення властивості Caption об'єкту кнопка (у полі Caption ввести потрібну назву об'єкта), яка подія може бути пов'язана з об'єктом кнопка (OnClick — при натисненні кнопки у полі напис з'являється відповідь, OnMouseMove — під час наведення вказівника миші на кнопку, з'являється відповідь тощо), яка властивість може бути застосована до об'єктів 5, 6, 7 або 60, -20, 30 (Visible — об'єкт невидимий).
3. Формі можна задати ім'я у полі Caption — «Правила безпечної поведінки в Інтернеті», змінити колір тла форми у полі Color — clGradientActiveCaption, змінити розміри вікна форми у полі Height — 500, Width — 500. На формі розмістити два написи (Label1 та Label2) з правилами та корисними посиланнями (підвести вказівник до зображення напису (TLabel) на панелі інструментів на вкладці Standard ⇒ два рази натиснути лівою клавішею миші (аналогічно зробити ще один напис) ⇒ у властивостях напису Label1 в полі Caption змінити текст за замовчуванням на правила безпечної поведінки в Інтернеті, Label2 — помістити текст з корисними посиланнями) ⇒ у полі Visible для Label1 та Label2 змінити значення на False ⇒ на формі розмістити дві кнопки (Button1 та Button2) (підвести вказівник до зображення кнопки на панелі інструментів на вкладці Standard ⇒ два рази натиснути лівою клавішею миші) ⇒ у властивостях кнопки в полі Caption змінити текст на Правила (Корисні посилання для Button2) ⇒ обираємо в меню Інспектор об'єктів вкладку Події ⇒ знаходимо поле OnClick та два рази натискаємо лівою клавішею миші ⇒ у поле праворуч від Інспектора об'єктів у вказаному курсором місці записуємо процедури, які мають команди:

```
Procedure TForm1.Button1Click(Sender:TObject);
```

```
begin  
label1.Visible:=True;  
label2.Visible:=False;  
end;
```

та відповідно друга кнопка,

```
Procedure TForm1.Button2Click(Sender:TObject);
```

```
begin  
label1.Visible:= False;  
label2.Visible:=True;  
end;
```

на панелі інструментів натиснути кнопку Запустити (або F9) ⇒ перевірити правильність виконання програми ⇒ зберегти проект (у вкладці Файл обрати ко-

манду Зберегти як ⇒ обрати місце збереження проекту ⇒ ввести ім'я проекту ⇒ натиснути Зберегти).

Працюємо самостійно

1. Завантажити Lazarus (Пуск ⇒ Всі програми ⇒ Lazarus) ⇒ у вкладці Проект обрати Створити проект ⇒ підвести вказівник миші до зображення напису (TLabel) на панелі інструментів на вкладці Standard ⇒ два рази натиснути ліву клавішу миші ⇒ у властивостях напису Label1 в полі Caption змінити текст за замовчуванням на «*» ⇒ використовуючи комбінації клавіш Ctrl+C — копіювати, Ctrl+V — вставити, створити підписи ще 14 написів ⇒ на формі розмістити ще дві кнопки (Button1 та Button2) (підвести вказівник до зображення кнопки на панелі інструментів на вкладці Standard ⇒ два рази натиснути лівою клавішею миші) ⇒ у властивостях кнопки в полі Caption змінити текст на 1 та 2 відповідно ⇒ обираємо в меню Інспектор об'єктів вкладку Події ⇒ знаходимо поле OnClick та два рази натискаємо лівою клавішею миші ⇒ у поле праворуч від Інспектора об'єктів у вказаній курсором місці записуємо процедури, які мають команди:

```
procedure TForm1.Button1Click(Sender:TObject);
```

```
begin
```

```
Label1.Visible:=False;
```

```
Label2.Visible:=True;
```

```
Label3.Visible:=False;
```

```
Label4.Visible:=True;
```

```
Label5.Visible:=True;
```

```
Label6.Visible:=False;
```

```
Label7.Visible:=False;
```

```
Label8.Visible:=True;
```

```
Label9.Visible:=False;
```

```
Label10.Visible:=False;
```

```
Label11.Visible:=True;
```

```
Label12.Visible:=False;
```

```
Label13.Visible:=False;
```

```
Label14.Visible:=True;
```

```
Label15.Visible:=False;
```

```
end;
```

та відповідно друга кнопка

```
procedure TForm1.Button2Click(Sender:TObject);
```

```
begin
```

```
Label1.Visible:=True;
```

```
Label2.Visible:=True;
```

```
Label3.Visible:=True;
```

```
Label4.Visible:=False;
```

```
Label5.Visible:=False;
```

```
Label6.Visible:=True;
```

```
Label7.Visible:=False;
```

```
Label8.Visible:=False;
```

```
Label9.Visible:= True;
```



```
Label10.Visible:=False;  
Label11.Visible:=True;  
Label12.Visible:=False;  
Label13.Visible:=True;  
Label14.Visible:=True;  
Label15.Visible:=True;  
end;
```

на панелі інструментів натиснути кнопку Запустити (або F9) ⇒ перевірити правильність виконання програми ⇒ зберегти проект (у вкладці Файл обрати команду Зберегти як ⇒ обрати місце збереження проекту ⇒ ввести ім'я проекту Табло ⇒ натиснути Зберегти).

2. Завантажити Lazarus (Пуск ⇒ Всі програми ⇒ Lazarus) ⇒ у вкладці Проект обрати Створити проект ⇒ підвести вказівник миші до зображення напису (TLabel) на панелі інструментів на вкладці Standard ⇒ два рази натиснути ліву клавішу миші ⇒ у властивостях напису Label1 в полі Caption змінити текст за замовчуванням на «Л» ⇒ використовуючи комбінації клавіш Ctrl+C — копіювати, Ctrl+V — вставити, створити підписи ще 2 написи (зі значеннями Caption Label2 — I та Label3 — C) ⇒ аналогічним чином створюємо ще 6 написів, але повністю видаляємо у полі Caption текст ⇒ для привабливого виду розміщуємо порожні написи один під одним (для цього можна скористатися властивостями Top та Left) ⇒ на формі розмістити кнопку (підвести вказівник до зображення кнопки на панелі інструментів на вкладці Standard ⇒ два рази натиснути лівою клавішею миші) ⇒ у властивостях кнопки в полі Caption змінити текст Розпочати ⇒ обираємо в меню Інспектор об'єктів вкладку Події ⇒ знаходимо поле OnClick та два рази натискаємо лівою клавішею миші ⇒ у поле праворуч від Інспектора об'єктів у вказаній курсором місці записуємо процедури, які мають команди:

```
procedure TForm1.Button1Click(Sender:TObject);  
begin  
Label4.Caption:=Label1.Caption+Label2.Caption+Label3.Caption;  
Label5.Visible:=Label2.Caption+Label3.Caption+Label1.Caption;  
Label6.Visible:=Label3.Caption+Label1.Caption+Label2.Caption;  
Label7.Visible:=Label3.Caption+Label2.Caption+Label1.Caption;  
Label8.Visible:=Label1.Caption+Label3.Caption+Label2.Caption;  
Label9.Visible:=Label2.Caption+Label1.Caption+Label3.Caption;  
end;
```

на панелі інструментів натиснути кнопку Запустити (або F9) ⇒ перевірити правильність виконання програми ⇒ зберегти проект (у вкладці Файл обрати команду Зберегти як ⇒ обрати місце збереження проекту ⇒ ввести ім'я проекту Табло ⇒ натиснути Зберегти).

4. Завантажити Lazarus (Пуск ⇒ Всі програми ⇒ Lazarus) ⇒ у вкладці Проект обрати Створити проект ⇒ зробити форму поточною ⇒ змінити значення властивостей Form1: Caption — Україна; Height — 200, Width — 300, Color — clMenuHighlight ⇒ розмістити на формі Form1 кнопку (підвести вказівник до зображення кнопки на панелі інструментів на вкладці Standard ⇒ два рази натиснути лівою клавішею миші) ⇒ змінити значення властивостей Form1: Caption —

Моя Україна; Height — 100, Width — 250, Font — за власним уподобанням ⇒ на панелі інструментів обрати кнопку Новий елемент: Форма ⇒ зробити форму поточною ⇒ змінити значення властивостей Form2: Caption — Опис; Height — 300, Width — 500, Color — cYellow ⇒ розмістити на формі Form2 три кнопки (підвести вказівник до зображення кнопки на панелі інструментів на вкладці Standard ⇒ два рази натиснути лівою клавішею миші) ⇒ змінити значення властивостей Button1: Caption — Столиця; Button2: Caption — Державний устрій; Button3: Caption — Територія та населення ⇒ розмістити на формі Form2 три написи (підвести вказівник до зображення кнопки на панелі інструментів на вкладці Standard ⇒ два рази натиснути лівою клавішею миші) ⇒ змінити значення властивостей Label1 у полі Caption відкрити діалогове вікно Діалог введення та ввести текст «Столиця: Київ. Географічна широта та довгота: 50°27' пн. ш., 30°30' сх. д.», Visible — False, Font — за власним уподобанням; Label2: у полі Caption відкрити діалогове вікно Діалог введення та ввести текст «Україна — парламентсько-президентська республіка. Незалежність проголошено: 24 серпня 1991 року», Visible — False, Font — за власним уподобанням; Label3: у полі Caption відкрити діалогове вікно Діалог введення та ввести текст «Кількість населення 42760500. Загальна площа території: 603628 км²», Visible — False, Font — за власним уподобанням ⇒ зробити Form1 поточною ⇒ у редакторі коду після слова «implementation» написати uses Unit2; (для підключення другої форми) ⇒ виділити кнопку Button1 на Form1 ⇒ обираємо в меню Інспектор об'єктів вкладку Події ⇒ знаходимо поле OnClick та два рази натискаємо лівою клавішею миші ⇒ у поле праворуч від Інспектора об'єктів у вказаній курсором місці записуємо процедури, які має команду:

```
Unit2.Form2.ShowModal;
```

⇒ зробити Form2 поточною ⇒ виділити кнопку Button1 на Form2 ⇒ обираємо в меню Інспектор об'єктів вкладку Події ⇒ знаходимо поле OnClick та два рази натискаємо лівою клавішею миші ⇒ у поле праворуч від Інспектора об'єктів у вказаній курсором місці записуємо процедури, які мають команди:

```
Label1.Visible:=True;  
Label2.Visible:= False;  
Label3.Visible:= False;
```

виділити кнопку Button2 на Form2 ⇒ обираємо в меню Інспектор об'єктів вкладку Події ⇒ знаходимо поле OnClick та два рази натискаємо лівою клавішею миші ⇒ у поле праворуч від Інспектора об'єктів у вказаній курсором місці записуємо процедури, які мають команди:

```
Label1.Visible:=False;  
Label2.Visible:=True;  
Label3.Visible:= False;
```

виділити кнопку Button3 на Form2 ⇒ обираємо в меню Інспектор об'єктів вкладку Події ⇒ знаходимо поле OnClick та два рази натискаємо лівою клавішею миші ⇒ у поле праворуч від Інспектора об'єктів у вказаній курсором місці записуємо процедури, які мають команди:

```
Label1.Visible:=False;  
Label2.Visible:= False;
```

Label3.Visible:= True;

на панелі інструментів натиснути кнопку Запустити (або F9) ⇒ перевірити правильність виконання програми ⇒ зберегти проект (у вкладці Файл обрати команду Зберегти як ⇒ обрати місце збереження проекту ⇒ ввести ім'я проекту Моя Україна ⇒ натиснути Зберегти).

РОЗДІЛ 6. АЛГОРИТМИ РОБОТИ З ОБ'ЄКТАМИ ТА ВЕЛИЧИНАМИ

20. Величини, їх типи та властивості

Обговорюємо

1. Для посилання на величину у виразах під час створення програм використовують імена величин. Позначення імен називають ідентифікаторами. Їх добирають у вигляді деякого скінченного впорядкованого набору літер і цифр, який починається з літери або символу підкреслення.
2. Константа — величина, яка має сталі значення в будь-які моменти часу.
3. Тип змінної визначає: яких значень може набувати ця змінна; які операції над нею можна виконувати; який обсяг оперативної пам'яті буде виділено для зберігання значення цієї змінної.
4. Імена змінних та їх типи вказуються в проекті після ключового слова var.

Працюємо в парах

1. Byte — 5, 30; Integer — 500, 9058; Smallint — 976, 186; Real — 11.3; 56.0; Int — 0.12456; Float — 25.7+e18, 0.999999.

Працюємо самостійно

1. а) Створення програми у середовищі PyCharm: завантажити середовище PyCharm (Пуск ⇒ Всі програми ⇒ JetBrains ⇒ JetBrains PyCharm 2016.1.4, який знаходиться на Робочому столі) ⇒ у меню File обрати команду New Project ⇒ у діалоговому вікні вікні Create Project, що відкрилося обрати папку збереження проекту та задати ім'я проекту «а» ⇒ натиснути кнопку Create ⇒ у меню File обрати команду New ⇒ у списку New File, що відкрився вибрати Python File ⇒ натиснути кнопку ОК ⇒ у вікні редактора коду ввести текст програми:

```
r=4
kr=int(input('введіть кількість рядів'))
k=r*kr
print(k)
```

У меню Run обрати команду Run ⇒ перевірити правильність виконання програми.

б) Створення програми у середовищі PyCharm: завантажити середовище PyCharm (Пуск ⇒ Всі програми ⇒ JetBrains ⇒ JetBrains PyCharm 2016.1.4, який знаходиться на Робочому столі) ⇒ у меню File обрати команду New Project ⇒ у діалоговому вікні вікні Create Project, що відкрилося обрати папку збереження проекту та задати ім'я проекту «б» ⇒ натиснути кнопку Create ⇒ у меню File

обрати команду New \Rightarrow у списку New File, що відкрився вибрати Python File \Rightarrow натиснути кнопку OK \Rightarrow у вікні редактора коду ввести текст програми:

```
r=4
kr=int(input('введіть кількість рядів'))
k=r*kr+2
print (k)
```

У меню Run обрати команду Run \Rightarrow перевірити правильність виконання програми.

в) Створення програми у середовищі PyCharm: завантажити середовище PyCharm (Пуск \Rightarrow Всі програми \Rightarrow JetBrains \Rightarrow JetBrains PyCharm 2016.1.4, який знаходиться на Робочому столі) \Rightarrow у меню File обрати команду New Project \Rightarrow у діалоговому вікні вікні Create Project, що відкрилося обрати папку збереження проекту та задати ім'я проекту «в» \Rightarrow натиснути кнопку Create \Rightarrow у меню File обрати команду New \Rightarrow у списку New File, що відкрився вибрати Python File \Rightarrow натиснути кнопку OK \Rightarrow у вікні редактора коду ввести текст програми:

```
r1=4
r2=4 / ? r2=5
kr1=int(input('введіть кількість рядів 1 класу'))
kr2=int(input('введіть кількість рядів 2 класу'))
k=r1*kr1+ r2*kr2+2
print (k)
```

У меню Run обрати команду Run \Rightarrow перевірити правильність виконання програми.

2. Створення програми у середовищі PyCharm: завантажити середовище PyCharm (Пуск \Rightarrow Всі програми \Rightarrow JetBrains \Rightarrow JetBrains PyCharm 2016.1.4, який знаходиться на Робочому столі) \Rightarrow у меню File обрати команду New Project \Rightarrow у діалоговому вікні вікні Create Project, що відкрилося обрати папку збереження проекту та задати ім'я проекту «Сімейний депозит» \Rightarrow натиснути кнопку Create \Rightarrow у меню File обрати команду New \Rightarrow у списку New File, що відкрився вибрати Python File \Rightarrow натиснути кнопку OK \Rightarrow у вікні редактора коду ввести текст програми:

```
p1=0.18
s1=int(input('введіть суму'))
s=sr1*0.18*100
print (s)
```

У меню Run обрати команду Run \Rightarrow перевірити правильність виконання програми.

3. Створення програми у середовищі PyCharm: завантажити середовище PyCharm (Пуск \Rightarrow Всі програми \Rightarrow JetBrains \Rightarrow JetBrains PyCharm 2016.1.4, який знаходиться на Робочому столі) \Rightarrow у меню File обрати команду New Project \Rightarrow у діалоговому вікні вікні Create Project, що відкрилося обрати папку збереження проекту та задати ім'я проекту «Електрика» \Rightarrow натиснути кнопку Create \Rightarrow у меню File обрати команду New \Rightarrow у списку New File, що відкрився вибрати Python File \Rightarrow натиснути кнопку OK \Rightarrow у вікні редактора коду ввести текст програми:

```
1) r1=int(input('введіть r1'))
```

```
r2:=int(input('введіть r2'))
r3:=int(input('введіть r3'))
r4:=int(input('введіть r4'))
r:=r1+r2+r3+r4
print (r)
```

У меню Run обрати команду Run ⇒ перевірити правильність виконання програми.

Аналогічно виконати для паралельного з'єднання:

```
2) r1:=int(input('введіть r1'))
r2:=int(input('введіть r2'))
r3:=int(input('введіть r3'))
r4:=int(input('введіть r4'))
r:=1/r1+1/r2+1/r3+1/r4
print (r)
```

У меню Run обрати команду Run ⇒ перевірити правильність виконання програми.

Завантажити Lazarus (Пуск ⇒ Всі програми ⇒ Lazarus) ⇒ у вкладці Проект обрати Створити проект ⇒ підвести вказівник миші до зображення поля (TEdit) на панелі інструментів на вкладці Standard ⇒ два рази натиснути ліву клавішу миші ⇒ у властивостях видалити в полі Text текст за замовчуванням (аналогічно зробити ще три поля: маса та ціна цукерок, маса та ціна печива) ⇒ підвести вказівник до зображення напису (TLabel) на панелі інструментів на вкладці Standard ⇒ два рази натиснути лівою клавішею миші ⇒ у властивостях кнопки в полі Caption змінити текст за замовчуванням ⇒ підвести вказівник до зображення кнопки на панелі інструментів на вкладці Standard ⇒ два рази натиснути лівою клавішею миші ⇒ у властивостях кнопки в полі Caption змінити текст на Отримати відповідь ⇒ обираємо в меню Інспектор об'єктів вкладку Події ⇒ знаходимо поле OnClick та два рази натискаємо лівою клавішею миші ⇒ у поле праворуч від Інспектора об'єктів у вказаній курсором місці записуємо процедури, які мають команди:

```
1) var r1, r2, r3, r4, r: real;
begin
r1:=StrToFloat(Edit1.Text);
r2:=StrToFloat(Edit2.Text);
r3:=StrToFloat(Edit3.Text);
r4:=StrToFloat(Edit4.Text);
r:=r1+r2+r3+r4;
Label1.Caption:=FloatToStr(r);
end;
```

зберегти проект (у вкладці Файл обрати команду Зберегти як ⇒ обрати місце збереження проекту ⇒ ввести ім'я проекту Електрика ⇒ натиснути Зберегти).

2) Аналогічно виконати для паралельного з'єднання

```
var r1, r2, r3, r4, r: real;
begin
r1:=StrToFloat(Edit1.Text);
```

```

r2:=StrToFloat(Edit2.Text);
r3:=StrToFloat(Edit3.Text);
r4:=StrToFloat(Edit4.Text);
r:=1/r1+1/r2+1/r3+1/r4;
Label1.Caption:=FloatToStr(r);
end;

```

зберегти проект (у вкладці Файл обрати команду Зберегти як ⇒ обрати місце збереження проекту ⇒ ввести ім'я проекту Електрика ⇒ натиснути Зберегти).

4. Завантажити Lazarus (Пуск ⇒ Всі програми ⇒ Lazarus) ⇒ у вкладці Проект обрати Створити проект ⇒ змінити значення властивостей Form1: Caption — Залишок; Height — 200, Width — 200; Color — clInactiveCaption ⇒ підвести вказівник миші до зображення напису (TLabel) на панелі інструментів на вкладці Standard ⇒ два рази натиснути лівою клавішею миші ⇒ у властивостях кнопки в полі Caption змінити текст за замовчуванням ⇒ підвести вказівник до зображення кнопки на панелі інструментів на вкладці Standard ⇒ два рази натиснути лівою клавішею миші ⇒ у властивостях кнопки в полі Caption змінити текст на Обчислити залишок тканини, властивість Font змінити за власним бажанням ⇒ обираємо в меню Інспектор об'єктів вкладку Події ⇒ знаходимо поле OnClick та два рази натискаємо лівою клавішею миші ⇒ у поле праворуч від Інспектора об'єктів у вказаній курсором місці записуємо процедури, які мають команди:

```

var a, R, S: real;
begin
a:=StrToInt(InputBox('Сторона квадрата', 'Введіть сторону квадрата:', ''));
R:=StrToInt(InputBox('Радіус кола', 'Введіть радіус кола:', ''));
S:=sqrt(a)-pi*sqrt(R);
Label1.Caption:=FloatToStr(S);
end;

```

зберегти проект (у вкладці Файл обрати команду Зберегти як ⇒ обрати місце збереження проекту ⇒ ввести ім'я проекту Залишок ⇒ натиснути Зберегти).

5. Створення програми у середовищі PyCharm: завантажити середовище PyCharm (Пуск ⇒ Всі програми ⇒ JetBrains ⇒ JetBrains PyCharm 2016.1.4, який знаходиться на Робочому столі) ⇒ у меню File обрати команду New Project ⇒ у діалоговому вікні вікні Create Project, що відкрилося обрати папку збереження проекту та задати ім'я проекту «Обчислення» ⇒ натиснути кнопку Create ⇒ у меню File обрати команду New ⇒ у списку New File, що відкрився вибрати Python File ⇒ натиснути кнопку OK ⇒ у вікні редактора коду ввести текст програми:

```

1) from math import fabs
x=int(input('введіть x'))
y=int(input('введіть y'))
z=(fabs(x)+ fabs(y))/(1+fabs(x*(y)))
print (z)
2) from math import fabs
x=int(input('введіть x'))
y=int(input('введіть y'))
z=1+(fabs(y-x)+(((y-x)**2)/2)+(((y-x)**3)/3)

```

```

print (z)
3) from math import fabs
x=int(input('введіть x'))
y=int(input('введіть y'))
z=2*y/cos(x-pi/6)
print (z)
4) from math import fabs
x=int(input('введіть x'))
y=int(input('введіть y'))
z=x/(1+(x*2/3+((2*x)**2)))
print (z)

```

У меню Run обрати команду Run ⇒ перевірити правильність виконання програми.

6. Завантажити Lazarus (Пуск ⇒ Всі програми ⇒ Lazarus) ⇒ у вкладці Проект обрати Створити проект ⇒ змінити значення властивостей Form1: Caption — Домашня бухгалтерія; Height — 400, Width — 300; Color — clMoneyGreen ⇒ підвести вказівник миші до зображення напису (TLabel) на панелі інструментів на вкладці Standard ⇒ два рази натиснути лівою клавішею миші ⇒ у властивостях кнопки в полі Caption змінити текст за замовчуванням ⇒ підвести вказівник до зображення кнопки на панелі інструментів на вкладці Standard ⇒ два рази натиснути лівою клавішею миші ⇒ у властивостях кнопки в полі Caption змінити текст на Обчислити залишок тканини, властивість Font змінити за власним бажанням ⇒ обираємо в меню Інспектор об'єктів вкладку Події ⇒ знаходимо поле OnClick та два рази натискаємо лівою клавішею миші ⇒ у поле праворуч від Інспектора об'єктів у вказаній курсором місці записуємо процедури, які мають команди:

```

var d1, d2, d3, v1, v2, v3, v4, v5, v: real;
begin
d1:=StrToInt(InputBox('Дохід мами, 'Введіть дохід мами', ''));
d2:=StrToInt(InputBox('Дохід тата, 'Введіть дохід тата', ''));
d3:=StrToInt(InputBox('Дохід брата, 'Введіть дохід брата', ''));
v1:=StrToInt(InputBox('Комунальні послуги, 'Введіть витрати на комунальні послуги', ''));
v2:=StrToInt(InputBox('Транспорт, 'Введіть витрати на транспорт', ''));
v3:=StrToInt(InputBox('Харчування, 'Введіть витрати на харчування', ''));
v4:=StrToInt(InputBox('Зв'язок, 'Введіть витрати на зв'язок,', ''));
v5:=StrToInt(InputBox('Інші витрати, 'Введіть загальні витрати на інші послуги', '0'));
v:=(d1+d2+d3)-(v1+v2+v3+v4+v5);
Label1.Caption:=FloatToStr(v);
end;

```

зберегти проект (у вкладці Файл обрати команду Зберегти як ⇒ обрати місце збереження проекту ⇒ ввести ім'я проекту Домашня бухгалтерія ⇒ натиснути Зберегти).

22. Текстові величини та операції над ними

Обговорюємо

2. Значенням величини символьного типу може бути один символ — літера, цифра чи знак. Набір символів утворює рядок.
3. Надавати значення текстовим величинам можна наступними способами: за допомогою оператора присвоєння; під час виконання програми, що містить команди або вікна введення; за допомогою елемента управління текстове поле.
4. Текстове поле створюється на екранній формі в середовищі Lazarus за допомогою компонента Edit.
6. Над текстовими величинами виконують операцію склеювання — поєднання кількох рядкових величин, яку позначають символом «+».
5. У програмі Python при роботі з текстовими величинами можна виконувати операцію зрізу, за допомогою якої копіюється послідовність або її частина.
6. У програмі Free Pascal виконують такі функції над текстовими величинами: визначення кількості символів у рядку; копіювання символу, видалення символу; вставлення одного рядка в інший; визначення номера позиції входження одного рядка в інший.

Працюємо в парах

1. Python:

I варіант

```
s=input('Введіть слово')
s2=s[11]+s[10]+s[9]+s[8]+ s[7]+ s[6]+ s[5]+ s[4]+ s[3]+ s[2]+ s[1]
print (s2)
```

II варіант

```
s=input('Введіть слово')
s2=s[: -1]
print (s2)
```

Free Pascal (потрібно підключити модуль Uses LCLProc);

```
var s1, s2: string;
```

```
begin
```

```
s1:=Edit1.Text;
```

```
s2:=UTF8copy(s1,11,1)+ UTF8copy(s1,10,1)+ UTF8copy(s1,9,1)+
```

```
UTF8copy(s1,8,1)+ UTF8copy(s1,7,1)+ UTF8copy(s1,6,1)+
```

```
UTF8copy(s1,5,1)+ UTF8copy(s1,4,1)+ UTF8copy(s1,3,1)+
```

```
UTF8copy(s1,2,1)+ UTF8copy(s1,1,1);
```

```
Edit2.Text:=s2;
```

```
end;
```

```
1) s1=input('Введіть змінну s1')
```

```
s2=s1[: 9]+''+ s1[33:44]+''+s1[20:33]+''+s1[9 : 20]
```

```
print(s2)
```

```
2) s1=input('Введіть змінну s1')
```

```
s2=s1[: 5]+''+ s1[5:7]+''+s[7:10]+''+ s1[10 : 16]+''+s1[16:21]
```

```
print(s2)
```

3. Free Pascal:

```
var pr, im, ba, z: string;
```



```

begin
pr:=Edit1.Text;
im:=Edit2.Text;
ba:=Edit3.Text;
z:=pr+' '+im+' '+ba+'', Вас зараховано до гуртка технічної творчості';
Label1.Caption:=z;
end;
Python:
pr=input('Вкажіть прізвище, ім'я, по батькові')
print(pr, ' ', Вас зараховано до гуртка технічної творчості')

```

Працюємо самостійно

- ```

1) s1=input('Введіть змінну s1')
s2=s1[4]+s1[10]+s1[11]+s1[7]+s1[3]+s1[9]
print(s2)
2) s1=input('Введіть змінну s1')
s2=s1[9]+s1[1]+s1[6]+s1[8]+s1[7]+s1[11]
print(s2)
3) s1=input('Введіть змінну s1')
s2=s1[3:6]+s1[2]+s1[6:8] ???/ з 3—5 і 6—7
print(s2)
4) s1=input('Введіть змінну s1')
s2=s1[7:9]+s1[5]+s1[4]+s1[3] ???/ з 7 по 8 ME
print(s2)

```
- Завантажити Lazarus (Пуск ⇒ Всі програми ⇒ Lazarus) ⇒ у вкладці Проект обрати Створити проект ⇒ змінити значення властивостей Form1: Caption — Розрядні одиниці; Height — 300, Width — 300; Color — clScrollBar ⇒ підвести вказівник миші до зображення поля (TEdit) на панелі інструментів на вкладці Standard ⇒ два рази натиснути лівою клавішею миші ⇒ аналогічним способом створити друге поле та у властивостях Enabled в полі поставити значення False ⇒ підвести вказівник до зображення кнопки на панелі інструментів на вкладці Standard ⇒ два рази натиснути лівою клавішею миші ⇒ у властивостях кнопки в полі Caption змінити текст на Розкласти, властивість Font змінити за власним бажанням ⇒ обираємо в меню Інспектор об'єктів вкладку Події ⇒ знаходимо поле OnClick та два рази натискаємо лівою клавішею миші ⇒ у поле праворуч від Інспектора об'єктів у вказаній курсором місці записуємо процедури, які мають команди:

```

var s1, s2: string;
begin
s1:=Edit1.Text;
s2:=s1[1]+'*1000+'+s1[2]+'*100+'+s1[3]+'*10+'+s1[4]+'*1';
Edit2.Text:=s2;
end;

```

зберегти проект (у вкладці Файл обрати команду Зберегти як ⇒ обрати місце збереження проекту ⇒ ввести ім'я проекту Розрядні одиниці ⇒ натиснути Зберегти).

4. Завантажити Lazarus (Пуск  $\Rightarrow$  Всі програми  $\Rightarrow$  Lazarus)  $\Rightarrow$  у вкладці Проект обрати Створити проект  $\Rightarrow$  підвести вказівник миші до зображення поля (TEdit) на панелі інструментів на вкладці Standard  $\Rightarrow$  два рази натиснути лівою клавішею миші  $\Rightarrow$  аналогічним способом створити ще два поля та у властивостях Enabled в полі Edit3 поставити значення False  $\Rightarrow$  підвести вказівник до зображення кнопки на панелі інструментів на вкладці Standard  $\Rightarrow$  два рази натиснути лівою клавішею миші  $\Rightarrow$  у властивостях кнопки в полі Caption змінити текст на +, -, \*, /; властивість Font змінити за власним бажанням  $\Rightarrow$  до кожної кнопки обираємо в меню Інспектор об'єктів вкладку Події  $\Rightarrow$  знаходимо поле OnClick та два рази натискаємо лівою клавішею миші  $\Rightarrow$  у поле праворуч від Інспектора об'єктів у вказаній курсором місці записуємо для кожної кнопки процедури, які мають команди:

```
procedure TForm1.Button1Click(Sender; TObject);
var ch1, ch2, rez: real
begin
 ch1:=StrToFloat(Edit1.Text);
 ch2:=StrToFloat(Edit2Text);
 rez:= ch1+ch2;
 Edit3.Caption:=FloatToStr(rez);
end;
```

```
procedure TForm1.Button2Click(Sender; TObject);
var ch1, ch2, rez: real
begin
 ch1:=StrToFloat(Edit1.Text);
 ch2:=StrToFloat(Edit2Text);
 rez:= ch1-ch2;
 Edit3.Caption:=FloatToStr(rez);
end;
```

```
procedure TForm1.Button3Click(Sender; TObject);
var ch1, ch2, rez: real
begin
 ch1:=StrToFloat(Edit1.Text);
 ch2:=StrToFloat(Edit2Text);
 rez:= ch1*ch2;
 Edit3.Caption:=FloatToStr(rez);
end;
```

```
procedure TForm1.Button4Click(Sender; TObject);
var ch1, ch2, rez: real
begin
 ch1:=StrToFloat(Edit1.Text);
 ch2:=StrToFloat(Edit2Text);
 rez:= ch1/ch2;
 Edit3.Caption:=FloatToStr(rez);
end;
```

зберегти проект (у вкладці Файл обрати команду Зберегти як  $\Rightarrow$  обрати місце збереження проекту  $\Rightarrow$  ввести ім'я проекту Калькулятор  $\Rightarrow$  натиснути Зберегти).

## 24. Опрацювання величин логічного типу. Команда розгалуження

### Обговорюємо

1. Логічним виразом називають вираз, який може набувати одне з двох значень: True або False. Висловлювання є прикладом логічного виразу. Якщо висловлювання істинне, то вважають, що значення відповідного логічного виразу дорівнює True. Якщо висловлювання хибне, то вважають, що значення відповідного логічного виразу дорівнює False.
2. Змінними логічного типу називають змінні, які можуть набувати одне з двох значень — True або False.
3. Логічний тип змінних позначається boolean. Наприклад, `var x,y: boolean`. Часто для змінних логічного типу використовуються команди `x:=true` або `x:=false`.
4. Заперечення логічного виразу  $x$  називають логічний вираз, значення якого дорівнює **true**, якщо значення логічного виразу  $x$  дорівнює **false**, і дорівнює **false**, якщо значення логічного виразу  $x$  дорівнює **true**.
5. Кон'юнкцією двох логічних виразів  $x$  і  $y$  називають логічний вираз, значення якого є **true**, якщо значення кожного з логічних виразів  $x$  і  $y$  є **true**; якщо значення хоча б одного з логічних виразів  $x$  або  $y$  є **false**, то значення кон'юнкції теж **false**.
6. Диз'юнкція двох логічних виразів  $x$  і  $y$  — це логічний вираз, значення якого **true**, якщо значення хоча б одного з логічних виразів  $x$  або  $y$  є **true**; якщо значення кожного з логічних виразів  $x$  і  $y$  є **false**, то значення диз'юнкції теж **false**.
7. Пріоритет виконання логічних операцій: спочатку виконуються операції заперечення, потім операції кон'юнкції, потім — операції диз'юнкції. Для змінення цього порядку виконання логічних операцій використовують дужки.

### Працюємо в парах

1.  $a+b < b+c$ ,  $a >= b$ ,  $a-b < > 0$ , 'біт'='біт', 'байт'<'біт'.
2. 

```
t=int(input('Введіть температуру води'))
if t<0: print('Твердий стан')
else if t>100: print('Газоподібний стан')
else: print ('Рідкий стан')
```

### Працюємо самостійно

1. Логічний вираз набуватиме значення, які подані в таблиці:

|   | Free Pascal | Python |
|---|-------------|--------|
| 1 | True        | False  |
| 2 | True        | False  |
| 3 | True        | False  |
| 4 | True        | True   |
| 5 | False       | False  |
| 6 | True        | False  |

3. 

```
if ComboBox1.Text='1'=true
```

```

then Edit1.Text:= 'початковий рівень';
if ComboBox1.Text='2'=true
then Edit1.Text:= 'початковий рівень';
if ComboBox1.Text='3'=true
then Edit1.Text:= 'початковий рівень';
if ComboBox1.Text='4'=true
then Edit1.Text:= 'середній рівень';
if ComboBox1.Text='5'=true
then Edit1.Text:= 'середній рівень';
if ComboBox1.Text='6'=true
then Edit1.Text:= 'середній рівень';
if ComboBox1.Text='7'=true
then Edit1.Text:= 'достатній рівень';
if ComboBox1.Text='8'=true
then Edit1.Text:= 'достатній рівень';
if ComboBox1.Text='9'=true
then Edit1.Text:= 'достатній рівень';
if ComboBox1.Text='10'=true
then Edit1.Text:= 'високий рівень';
if ComboBox1.Text='11'=true
then Edit1.Text:= 'високий рівень';
if ComboBox1.Text='12'=true
then Edit1.Text:= 'високий рівень';

```

4.

```

var x, y: real;
begin
x:=StrToFloat(Edit1.Text);
y:= StrToFloat(Edit2.Text);
if ((x>=-2)) and (x<=2) and ((y>=-2) and (y<=2)) then
Edit3.Text:='Вітаємо! Влучний постріл'
else Edit3.Text:='Шкода, не влучив! ';
end;

```

а для другого варіанту?  $y \leq -x+2$  and  $y \geq -x-2$  and  $y \leq x+2$  and  $y \geq x-2$

6.

```

implementation
uses Unit2, Unit3, Unit4, Unit5, Unit6, Unit7, Unit8;
{$R*.lfm}
{TForm1}
begin
if ChekBox1.Checked then Form2.Show // червоний колір форми
else
if ChekBox2.Checked then Form3.Show // помаранчевий колір форми
else
if ChekBox3.Checked then Form4.Show // жовтий колір форми
else
if ChekBox4.Checked then Form5.Show // зелений колір форми

```

```

else
if ChekBox5.Checked then Form6.Show // блакитний колір форми
else
if ChekBox6.Checked then Form7.Show // синій колір форми
else
if ChekBox7.Checked then Form8.Show // фіолетовий колір форми
end;

```

## 25. Реалізація алгоритмів повторення мовами програмування

### Обговорюємо

1.

Блок-схема повного розгалуження. Особливості виконання повного розгалуження: обчислюється значення логічного виразу, якщо це значення істинне (true), то виконується послідовність команд 1 і після цього виконується команда, наступна за розгалуженням; якщо це значення хибне (false), то виконується послідовність команд 2 і після цього виконується команда, наступна за розгалуженням.

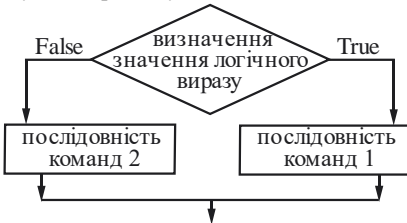


Рис. 25.1. Повне розгалуження

2.

Блок-схема неповного розгалуження. Особливості виконання повного розгалуження: обчислюється значення логічного виразу, якщо це значення істинне (true), то виконується послідовність команд і після цього виконується команда, наступна за розгалуженням; якщо це значення хибне (false), то виконується команда, наступна за розгалуженням.



Рис. 25.2. Неповне розгалуження

3. Відмінності виконання повного і неповного розгалуження: 1) обчислюється значення логічного виразу, якщо це значення істинне (true), то у **повному** розгалуженні виконується послідовність команд і після цього виконується команда, наступна за розгалуженням; 2) у **неповному** — виконується послідовність команд і після цього виконується команда, наступна за розгалуженням; якщо це значення хибне (false), то у **повному** розгалуженні виконується послідовність команд 2 і після цього виконується команда, наступна за розгалуженням, а у **неповному** — відразу виконується команда, наступна за розгалуженням.

4. Команда повного розгалуження у Free Pascal:

```

If <логічний вираз>
Then begin
<послідовність команд 1>
end
Else begin
<послідовність команд 2>

```

- end;
- Команда неповного розгалуження у Free Pascal:
 

```
If <логічний вираз>
Then begin
<послідовність команд>
end;
```
  - Відмінність виконання команд повного і неповного розгалуження у Object Pascal: 1) обчислюється значення логічного виразу, якщо це значення істинне (true), то у **повному** розгалуженні виконується послідовність команд і після цього виконується команда, наступна за розгалуженням; 2) у **неповному** — виконується послідовність команд і після цього виконується команда, наступна за розгалуженням; якщо це значення хибне (false), то у **повному** розгалуженні виконується послідовність команд 2 і після цього виконується команда, наступна за розгалуженням, а у **неповному** — відразу виконується команда, наступна за розгалуженням.
  - Цикл в алгоритмі — це будь-яка багатократно виконувана послідовність команд, організована певним чином, що міститься між операторними дужками.

### Працюємо в парах

#### 1. Узагальнення опису циклів з повторенням

|                                                               |                                                                          |                                    |        |
|---------------------------------------------------------------|--------------------------------------------------------------------------|------------------------------------|--------|
| Free Pascal                                                   | Цикл з передумовою                                                       |                                    | Python |
|                                                               | while <логічний вираз> do <команда>;                                     | while <логічний вираз> : <команда> |        |
|                                                               | ↓                                                                        |                                    |        |
|                                                               | Цикл з лічильником                                                       |                                    |        |
|                                                               | for i:<in> to <ik> do <команда>;<br>for i:<in> downto <ik> do <команда>; | for ш in <діапазон> : <команда>    |        |
|                                                               | ↓                                                                        |                                    |        |
| Цикл з післяумовою                                            |                                                                          |                                    |        |
| repeat <команда>;<br>until <логічний вираз припинення циклу>; |                                                                          |                                    |        |

### Працюємо самостійно

- 1 — 1) логічний вираз дорівнює False, тому цикл завершується,  
2)  $x=3, i=5$ ; 3)  $x=1, i=5$ ;  
2 — 1)  $s=35, n=11$ , 2)  $s=40, n=12, s=49, n=14$ .

## РОЗДІЛ 7. ТЕХНОЛОГІЇ ОПРАЦЮВАННЯ ЧИСЛОВИХ ДАНИХ У СЕРЕДОВИЩІ ТАБЛИЧНОГО ПРОЦЕСОРА

### 29. Обчислення в електронних таблицях

#### Обговорюємо

1. У формулах в електронних таблицях можна використовувати різні види посилань на клітинки: відносні (модифікуються під час копіювання формули), абсолютні (не модифікуються під час копіювання формули), мішані (модифікуються частково).
2. На клітинку B2 посилання матимуть такий вид: відносне — B2; абсолютне — \$B\$2; мішане — \$B2 або B\$2.
3. Відносні посилання модифікуються під час копіювання формули, абсолютні — не модифікуються під час копіювання формули, а мішані — модифікуються частково.
4. Ім'я клітинки належить до відносного типу посилань. Але якщо у формулі для посилання на клітинку використати її ім'я, то під час копіювання цієї формули це посилання не зміниться, воно вважається абсолютним.
5. Сторінка електронної таблиці має такі властивості: орієнтація аркуша друку, масштаб, розмір аркуша паперу для друку, якість друку, номер першої сторінки (на вкладці Сторінка); розміри полів, колонтитулів, центрування таблиці (на вкладці Поля); область друку, друк наскрізних рядків і стовпців, порядок друку сторінок (на вкладці Аркуш).

#### Працюємо в парах

1. При використанні відносного посилання на клітинку в формулі, в разі її копіювання посилання змінюється автоматично. Абсолютні посилання на комірку залишається незмінним, не залежно від копіювання формули в інше місце. Відповідно комбіновані (мішані) посилання змінюються частково. Якщо посилання абсолютне за стовпцем та відносне за рядком, то в разі копіювання посилання за стовпцем залишається незмінним, а рядок міняється в залежності від місця перенесення формули. Відповідно посилання, яке відносне за стовпцем та абсолютне за рядком навпаки: незмінним залишається в формулі рядок та змінюється стовець.
2. Гра «Аргументи функції»:

| Функція                                                                     | Можливі аргументи                                                                | Результат            |
|-----------------------------------------------------------------------------|----------------------------------------------------------------------------------|----------------------|
| ABS — повертає модуль (абсолютне значення) числа (число без знаку)          | =ABS (17)<br>математичний опис ( 17 )<br>=ABS (-17)<br>математичний опис ( -17 ) | 17<br>17             |
| ROUND — заокруглює число до вказаної кількості розрядів                     | =ROUND (18,1426; 2)                                                              | 18,14                |
| PI — повертає математичну константу «пі», округлену до 15 знаків після коми | =PI () — не містить аргументів                                                   | 3,1415926535897<br>9 |

|                                                        |                              |    |
|--------------------------------------------------------|------------------------------|----|
| POWER — повертає результат піднесення числа до степеня | =POWER (5; 2)                | 25 |
| SQRT — повертає додатне значення квадратного кореня    | =SQRT (25)                   | 5  |
| AVERAGE — повертає середнє (арифметичне) аргументів    | =AVERAGE (2; 3; 3; 5; 7; 10) | 5  |
| SUM — додає значення вказаних аргументів               | =SUM (2; 3; 3; 5; 7; 10)     | 30 |
| MAX — повертає найбільше число з набору значень        | =MAX (2; 6; 1; 9)            | 9  |
| MIN — повертає найменше число з набору значень         | =MIN (2; 6; 1; 9)            | 1  |

3. Значення в клітинці D2 буде набувати наступних значень (за умови, що клітинка B2=20, C2=10):

1) Відповідь: 20. (У формулі (=IF (B2<0; -B2; B2)) перевіряється, чи значення клітинки B2 менше 0, якщо умова істинна, то виводиться значення B2 зі знаком мінус, якщо хибна, то саме значення B2; в цьому випадку умова не виконується, тому виводиться значення B2).

2) Відповідь: більше. (У формулі (=IF (B2>C2; «більше»; «»)) перевіряється умова, що значення клітинці B2 більше за значення в клітинці C2, оскільки умова виконується, то відповідно виводиться «Значення\_якщо\_істина»).

3) Відповідь: 30. (У формулі (=IF (C2>0; B2+C2; B2-C2)) перевіряється умова, що значення клітинці C2 більше за значення в клітинці B2 (більше 0?), оскільки умова виконується, то відповідно виводиться «Значення\_якщо\_істина» «B2+C2»).

4. Значення логічних функцій (якщо в клітинці A1 = 100; B1 = 10):

1) =AND (A1<0; B1>0) — відповідь: False. (Функція AND перевіряє, чи всі аргументи істинні, якщо це так, то повертає значення True).

2) =OR (A1<0; B1>0) — відповідь: True. (Функція OR перевіряє, чи має хоча б один аргумент значення True і повертає значення True; значення False повертає в тому випадку, якщо всі аргументи False).

3) =AND (A1>20; B1<=50) — відповідь: True.

4) =NOT (A1<>0) — відповідь: False. (Функція NOT змінює значення True на False, а False на True).

5) =AND (A1>50; A1<>70; B1=10) — відповідь: True. / не було

6) =OR (A1<200; B1<>0; B1<20) — відповідь: True.

## 32. Діаграми різних типів. Друкування електронної таблиці

### Обговорюємо

1. Встановлення значень параметрів сторінки електронної таблиці для друку здійснюється на Стрічці елементами керування групи Параметри сторінки на вкладці



Розмітка сторінки або на вкладках діалогового вікна Параметри сторінки цієї групи.

2. Попередній перегляд електронної таблиці використовують для того, щоб переглянути, який вигляд матиме електронна таблиця на папері. У даному режимі буде відображено таблицю, підготовлену до друку, яка розділена на сторінки.
3. У режимі Попередній перегляд можна переглянути сторінки по одній, а в режимі Розмітка сторінки — одразу бачити кілька сторінок, розділених штриховими лініями. Переміщуючи штрихові лінії, можна змінювати розподіл вмісту по сторінках.
4. Для швидкого друку електронної таблиці треба виконати Office ⇒ Друк ⇒ Швидкий друк (зі значеннями властивостей сторінки за замовчуванням).
5. Для звичайного друку електронної таблиці виконують вказівки Office ⇒ Друк ⇒ Друк, у вікні Друк обираємо доступний принтер, кількість копій, об'єкт для друку (виділений діапазон, виділені аркуші чи всю книгу).

### **Працюємо в парах**

1. За результатами аналізу виявлено, що найчастіше використовуються такі типи діаграм, як стовпчаста або гістограма, далі — лінійна і кругова. Вони найбільше підходять для візуалізації зміни показників за певний період, відображення кількісних та якісних значень, аналізу зміни певних параметрів у часі.
2. Для відображення візуалізації певних кількісних або якісних показників за певний проміжок часу найчастіше використовуються стовпчасті, лінійна або кругові діаграми.
3. Знайдений мною «графік» є стовпчастою діаграмою курсу валют сьогоднішнього дня. Його можна замінити на лінійну діаграму, щоб проаналізувати зміни курсу тої чи іншої валюти впродовж усього дня.
4. «Назва діаграми», «Легенда», «Підписи даних» дають можливість відразу зрозуміти зміст інформації та полегшують сприйняття / читання всієї діаграми.
5. Запитання за даними діаграми (Рис. 32.29): «Яка кількість учнів була в 2010 (2011, 2012, 2013, 2014, 2015) році?»; «В якому році була найбільша (найменша) кількість хлопців (дівчат)?»; «Коли чисельність дівчат і хлопців була однаковою?».
7. Вибір типу діаграм для оформлення шкільної газети залежить від даних, які будуть представлені в діаграмі: 1) гістограма (секторна); 2) лінійна; 3) кільцева; 4) гістограма; 5) стовпчаста; 6) секторна (розрізана секторна діаграма).

### **Працюємо самостійно**

1. 1) Аналіз діаграми (Рис. 32.30): виділені екранізовані комікси вважають найуспішнішими, оскільки показники касових зборів склали більше, ніж 400 млн. дол.  
2) За втраченим бюджетом найбільша кількість грошей була витрачена на екранізацію «Люди Ікс: Остання Битва», найменша — «Бетмен», 1989.  
3) Чи відповідає дійсності гіпотеза:  
— Ні (про це свідчить показник «Люди Ікс: Остання Битва»);  
— Ні (не відповідають показники 2002 р. — «Людина-павук» та 2003 р. — «Люди Ікс-2»);  
— Ні (показники не відповідають твердженню — «Людина-павук» та «Людина-павук-2»);

— Так (відповідно до представлених фільмів на діаграмі, бюджет других версій більший);

— Так (збори від екранізації других версій менші).

4) Грошові збори від екранізації коміксів перевищують витрачений на них бюджет в:

— «Людина-павук», 2002 р. — у 5,9 разів;

— «Людина-павук-2», 2004 р. — у 3,9 рази;

— «Люди Ікс: Остання Битва», 2006 р. — у 2,08 рази;

— «Бетмен», 1989 р. — у 11,7 рази;

— «Люди Ікс-2», 2003 р. — у 3,7 рази.

5) Найбільші грошові збори отримано від екранізації фільму «Людина-павук», 2002 р.

### **33. Упорядкування та фільтрація даних у таблицях.**

#### ***Проміжні підсумки***

#### **Обговорюємо**

2. Сортування значень в Excel полягає у швидкому впорядкуванні даних для їх покращеного сприйняття та максимально ефективній організації. Дані в Excel можна сортувати:

— за текстом (від «А» до «Я» або від «Я» до «А»);

— числами (від найменшого до найбільшого та від найбільшого до найменшого);

— датою та часом (від найстаршого до наймолодшого, від наймолодшого до найстаршого) в одному або кількох стовпцях;

— за налаштуваннями списку;

— за форматом;

— за кольором клітинки;

— за кольором шрифту;

— набором піктограм.

Сортувати можна за рядками і стовпцями, останнє використовується частіше.

2. Для відсортування даних в алфавітно-цифровому порядку — текст, числа, дата і час — за зростанням, необхідно вибрати кнопку Сортування від «А» до «Я».

3. Для проведення швидкого сортування використовують кнопки Сортування від «А» до «Я» або від «Я» до «А» на вкладці Дані у групі Сортування й фільтр.

4. Сортування від найменшого до найбільшого змінюється на Сортування від А до Я, якщо сортуються дані текстового формату.

5. Щоб відсортувати рядки у виділеному діапазоні клітинок за значеннями в кількох стовпцях, треба: 1) виділити потрібний зв'язний діапазон клітинок; 2) виконати Дані ⇒ Сортування й фільтр ⇒ Сортувати; 3) встановити або змінити значення параметрів сортування: у списку поля Сортувати за вибрати ім'я першого стовпця; 4) визначити умови сортування за значеннями в іншому стовпці, вибравши кнопку Додати рівень або Копіювати рівень, встановити в полях рядка Потім за значення параметрів сортування за значеннями у другому вибраному стовпці; 5) за потребу повторити попередню команду; 6) якщо у виділений діапазон увійшов рядок заголовка, але позначку прапорця Мої дані мають заголовки не встановлено, встановити її, далі ОК.

6. Фільтрування — це вибір значень у клітинках електронної таблиці, що відповідають певним умовам. Після фільтрування в таблиці відображаються ті рядки, дані яких відповідають умовам фільтрування, інші — тимчасово приховані.
7. Фільтрування **числових** даних: спосіб 1) у списку чисел виділити або зняти виділення з одного або декількох чисел, за якими потрібно виконати фільтрування; спосіб 2) у великому за обсягом списку необхідно зняти виділення з пункту Виділити все та вибрати конкретні числа для фільтрування; спосіб 3) для задання критерію навести вказівник миші на пункт Фільтри чисел та вибрати одну з команд операторів порівняння або пункт **Користувацький фільтр**. У діалоговому вікні Користувацький авто фільтр ввести у полі число або кілька чисел.  
Фільтрування **текстових** даних: 1) у списку текстових значень виділити чи зняти виділення з одного або кількох текстових значень, за якими потрібно виконати фільтрування; 2) у великому списку, необхідно зняти виділення з пункту Виділити все та вибрати конкретні текстові значення для фільтрування; 3) для створення критерію необхідно навести вказівник миші на пункт Текстові фільтри та вибрати одну з команд оператора порівняння або вибрати пункт Користувацький фільтр. Наприклад, для фільтрування тексту, який починається з певного символу, необхідно обрати вказівку **Починається з**, а для тексту, який містить в будь-якому місці певні символи, вибрати пункт **Містить**.  
Користувацький фільтр дозволяє фільтрувати текст, який починається з певної букви або, який містить певний символ. Також при фільтруванні текстових даних можна використовувати символи узагальнення: ? (знак питання), \*(зірочка), - (тильда).
8. Розширений фільтр використовуються тоді, коли дані, які потрібно відфільтрувати, вимагають складних умов.
9. Умови для Розширеного фільтру задаються у відповідному діалоговому вікні (його можна викликати, перейшовши на вкладку **Дані** у групі **Сортування й фільтр**, команда **Додатково**).

## Працюємо в парах

1. Впорядкування даних використовується для забезпечення наступних вимог: легшого сприйняття інформації; зручного опрацювання даних; групування / об'єднання за певними властивостями або параметрами; швидкого пошуку необхідної інформації; відображення / відбору тільки найважливіших даних.
2. У побуті фільтри використовують для очищення води (від солей, важких металів, канцерогенів, дрібних частинок піску, хлору); у пирососах (для очищення повітря на виході від пилу); фільтри використовують в автомобілях (повітряні, паливні, масляні).  
Дія фільтрів у побуті та в табличному процесорі опосередковано подібна: очистити від зайвого, забезпечити отримання чогось в прийнятній формі, затримати вхід певних елементів.