

ОЧНИЙ ЕТАП ВСЕУКРАЇНСЬКОГО ДИСТАНЦІЙНОГО ТУРНІРУ З ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ 2014–2015 НАВЧАЛЬНОГО РОКУ: РЕКОМЕНДАЦІЇ ЩОДО РОЗВ'ЯЗУВАННЯ ЗАВДАНЬ

Потієнко В.О.,

учитель інформатики УФМЛ КНУ ім. Тараса Шевченка, кандидат педагогічних наук.

ЗАВДАННЯ 1. БАЗИ ДАНИХ

Як видно з умови завдання, робота з базою даних (БД) починається у вікні входу — вікні головної форми. Головна кнопкова форма створюється для зручної навігації по базі даних — виконує роль головного меню БД. Елементами запуску головної кнопкової форми є об'єкти форм і звітів. За завданням необхідно створити головну форму та інші, які б запускались клацанням мишею на відповідних їм кнопках.

Розглянемо процес виконання завдання: має сенс створити форми і запити за умовою завдання, а вже надалі перейти до створення головної кнопкової форми.

Створення форми реєстрації абітурієнтів

Проаналізуємо, як створити форму (бланк) реєстрації абітурієнтів: що має містити вказана форма, які процеси відбуваються після її закриття. Реєстрація, тобто введення відомостей про нового абітурієнта, означає додавання даних до таблиці. Але форми створюють на основі таблиць, отже, спочатку створюють структуру таблиці (з назвою **Абітурієнти**) з полями, даними в умові для реєстрації абітурієнта, а вже вигляд форми **Реєстрація абітурієнта** створюють на основі даних полів таблиці. Заповнення таблиці конкретними даними відбувається заповненням значеннями полів форми.

Проаналізуємо, які поля доцільно створювати в новій таблиці.

Недоцільно вводити до нової таблиці детальні дані ЗНО, оскільки в сертифікатах учнів повторюються назви предметів, а ЗНО з одного предмету здають багато учнів — тобто таблиця **Абітурієнти** і таблиця **Предмети** мають тип зв'язку багато – до – багатьох, який і реалізується проміжною таблицею **ЗНО** (через поле **Номер сертифікату** здійснюють зв'язок з таблицею **Абітурієнти**, а через поле **Предмет** — з таблицею **Предмет**). Отже, особисті дані абітурієнта будуть в таблиці **Абітурієнти**, подробиці складання ЗНО — у таблиці **ЗНО**, а форму **Реєстрація абітурієнтів** створюють на основі цих таблиць і таблиці **Предмети**.

Розглянемо більш детально властивості полів вказаних таблиць: для деяких із них застосовують шаблон (маску) введення значень, для деяких необхідно ввести умову на обмеження значень.

Маски введення складаються з одного обов'язкового і двох необов'язкових компонентів, розділених «;». Перший компонент є обов'язковим — це знак або рядок (послідовність знаків), вміст якого (заповнювачі) і визначає формат введення даних. Другий і третій компоненти не обов'язково вказувати; вони визначають спосіб зберігання вбудованих знаків маски в поле і знаки як заповнювач відповідно. У цьому завданні достатньо вказати перший компонент.

Наведемо приклади значень властивості маски введення стосовно полів з умови завдання:

- поля: **Прізвище**, **Ім'я**, **По батькові** — маска введення «перша літера велика, інші — малі (не більше 30

символів кожне)» — `>L<??????(29“?”)` (знаки «>» і «<» означають, що всі наступні знаки будуть введені у верхній регістр або у нижній відповідно, «L» — користувач має обов'язково ввести букву, а «?» означає, що користувач може ввести букву);

- поле **Серія атестата** — маска введення: рядок з двох літер — `LL`;

- поле **Номер атестата** — маска введення: рядок з 8 цифр — `00000000` (знак «0» означає обов'язкове введення цифри від 0 до 9, а «9» в масці означає, що користувач може ввести цифру).

Маску введення прописують безпосередньо як значення властивості або активізують властивість «маска» і вказують шаблон у вікні, що відкриється.

Маски введення недостатньо для коректного введення значень для деяких полів. У цьому випадку звертаються до властивості **Умова на значення**, у якій є можливість звернення до стандартних функцій. Наприклад, до властивості маски поля **Серія атестата** (умова — рядок з двох літер), варто додати ще одну властивість — умову, що визначить довжину введеного тексту (результат обчислення значення стандартної функції `Len` (текстовий тип) має дорівнювати 2).

Вигляд умовного виразу: `Len([Серія атестата])=2`.

Для поля **Номер атестата**: `Len([Номер атестата])=8`.

Для поля **Середній бал**: `>=1 And <=12`. (Для поля **Середній бал** також встановлюють властивість **Шкала 1**, оскільки значення в цьому полі — дійсні числа і після десяткової коми має бути 1 знак.)

Для введення адреси поштової скриньки:

`Is Null Or ((Like “*?@*.*”) And (Not Like “*[,]”))`.

Останнє пояснимо більш детально. Адреси електронної пошти можуть бути різної довжини, тому до них не підходять маски введення — визначають властивість **Умова на значення**. Наведена умова вказує, що адреса електронної пошти буде складатися з одного або декількох знаків, знака «@», ще одного або декількох знаків, крапки і ще одного або декількох знаків. Якщо ввести адресу електронної пошти, яка не відповідає умові на значення, введені дані будуть відхилені і з'явиться повідомлення, вказане у властивості **Повідомлення про помилку**. У цьому завданні про це нічого не сказано. Якщо властивість **Повідомлення про помилку** не задано, буде виведено стандартне повідомлення. (Введення умов на значення полів та виведення повідомлення про помилку з конкретним текстом повторюється в завданні кілька разів для решти форм та запитів).

Властивості поля пароль редагують у формі, оскільки записи таблиці заповнюють у формі.

Аналогічно полю **Середній бал** таблиці **Абітурієнти** встановлюють властивості поля **Результат** (кількість балів) у таблиці **ЗНО** з урахуванням умови завдання «число від 100 до 200, не більше однієї цифри після коми».

Перш ніж перейти до створення форм, а їх вигляд базується на полях кількох таблиць, створюють зв'язки між таблицями у вікні **Схема даних**.

Таблиці **ЗНО** і **Спеціальності** мають зв'язок через таблицю з умови **Спец-предмет**, яка, у свою чергу, реалізує зв'язок багато – до – багатьох між таблицями з умови **Спеціальності** й **Предмети**. У таблиці **Абітурієнти** також відсутні відомості щодо вибору спеціальності — вибір реалізовано через заявку на певну спеціальність: таблиця з особистими відомостями про абітурієнта має зв'язок з таблицею **Спеціальності** (дана в умові) через створену нову таблицю **Заявки**.

Приклад готової схеми даних для цього завдання наведено на рис. 1 (у правій частині 3 таблиці, надані умовою, а в лівій — нові, створені в процесі виконання завдання).

Отже, після створення таблиць **Абітурієнти** і **ЗНО** та встановлення зв'язків між ними, з таблицею **Предмети**, переходять до створення форми **Реєстрація абітурієнта**. Форма містить такі поля: усі поля таблиці **Абітурієнти**, поля **Рік складання** і **Результат** таблиці **ЗНО** та поле **Предмет** таблиці **Предмети**. У результаті отримують **Головну форму** **Абітурієнти** і підлеглу **ЗНО**.

Надалі в режим конструктора редагують форму у вікні властивостей: **КОНСТРУКТОР** — **Страница свойств**. Якщо поле форми виділене, відкриється вікно його властивостей, інакше у вікні, що відкриється, з випадаючого списку обирають необхідне поле.

Редагування зовнішнього вигляду форми згідно умови завдання краще здійснювати в режимі макета (вставляють зображення герба; напис у формі **Абітурієнти** замінюють на **Реєстрація абітурієнтів** тощо).

Щоб згенерувати пароль, звертаються до властивостей поля і на вкладці **Дані для поля** вводять

формулу за правилами введення формул у комірки електронних таблиць. «Пароль — автоматично згенерований рядок із шести випадкових символів ASCII-таблиці з номерами від 48 до 122» — сказано в умові (такими символами є цифри від 0 до 9, букви латинського алфавіту: великі й маленькі, та знаки пунктуації; вигляд таблиці додається до умови завдання). Кожний з 6-ти символів генерують випадково: його випадковість визначається випадковістю місця (порядкового номера) розташування в таблиці, а для визначення паролю створюють рядок, у якому використовують конкатенацію (&) 6-ти символів. Порядковий номер таблиці — величина числового типу, а функція, яка повертає символ на певному місці, має вигляд **Chr** (аргумент); тобто аргументом є порядкове місце в таблиці. Отже, завдання з випадкового встановлення паролю зводиться до завдання випадкової генерації числа від 48 до 122 (саме на цих місцях таблиці розташовані дозволені символи паролю) і застосування цього числа як аргумента функції **Chr**:

$$=Chr(Int(((122-48+1)*Rnd())+48))\& Chr(Int(((122-48+1)*Rnd())+48))\& ...$$

Наступний пункт виконання завдання — форматування полів даних **ЗНО**. Особливістю поля **Предмет** є не безпосереднє введення даних, а вибір з випадаючого списку. З цією метою в режимі конструктора редагують область значень **Предмет** — з контекстного меню для значень вказаного поля вибирають команду **Преобразовать элемент в — Поле со списком**. Предмети в списку є значеннями поля **Назва** таблиці **Предмет**. Працюють на вкладці **Данные**. Властивість **Данные** — слово **Предмет**. Необхідно визначити властивість **Источник строк** — із списку вибирають таблицю **Предмет** та активізують кнопку з «...»: одразу відкривається бланк побудови запитів. Запит ство-

рюють на основі таблиці **Предмет** з вибором її полів. Після того, як закрили і зберегли запит, властивість **Источник строк** змінилась на результат відпрацювання запиту. Властивістю **Тип источника строк** є **Таблица** или **Запрос**, а властивість **Присоединенный столбец** вказує на кількість під'єднаних стовпців, у даному випадку дорівнює 1.

Особливістю описаної форми є те, що в ній не відтворюються дані таблиці, а в разі запуску її бланк кожного разу пустий і, для того щоб додати дані з форми в таблицю, потрібна кнопка (за умовою це кнопка **Зберегти**), після клацання на якій виконувалася б необхідна подія. Але в цьому випадку має відбуватися не одна подія (наприклад, запам'ятати введені дані в таблиці), а кілька: відповідну таблицю доповнити даними; закрити форму; відкрити головну форму. Тобто необхідно створити макрос з вищезазначеними командами, а після клацання на кнопці **Зберегти макрос** запуститься в роботу.

Створення головної форми

Розглянемо, як створити вигляд головної форми відповідно до зразка, розмістити на ній кнопку **Зареєструватися**, після клацання на якій запускатиметься відповідна форма.

На стрічці **Работа с базами данных** має бути група **Диспетчер кнопочных форм**. Для її встановлення користуються такою послідовністю команд:

- **Файл/Параметры;**
- вибрати **Настройка ленты;**
- із списку **Настройка ленты** вибрати **Основные вкладки;**
- у вікні **Основные вкладок** вибрати стрічку **Работа с базами данных**, розкрити її;
- вибрати **Администратор**, розкрити список;
- виділити **Диспетчер компьютерных форм** і активізувати кнопку **Создать группу;**
- завершити, як і завжди, — кнопка **ОК**.

У відповідь на запуск команди **Работа с базами данных** — **Диспетчер кнопочных форм**, відкривається вікно з повідомленням про відсутність кнопочної форми і пропонується її створити («Да»). Як вищезазначено, головна кнопочна форма містить головну сторінку і підлеглі, які відкриваються з головної. Тому одразу пропонується створення головної сторінки. У вікні, що відкрилось:

- вибирають команду **Изменить;**

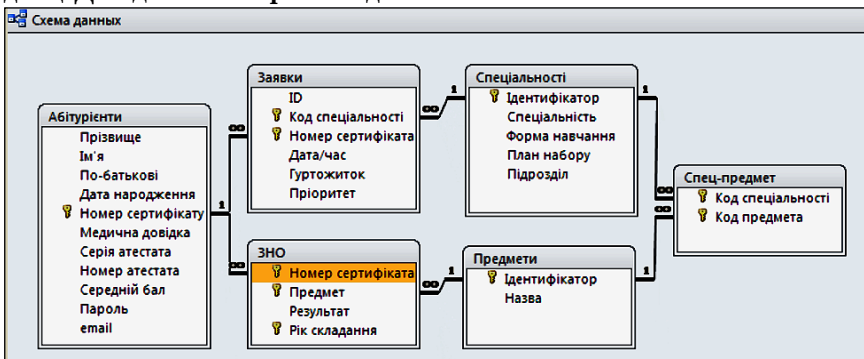


Рис. 1. Приклад зв'язків між таблицями у вікні **Схеми даних**

- у новому вікні у відповідний рядок вводять заголовок форми: **Вхід в систему**;
- вибирають команду **Создать**;
- заповнюють поля вікна відповідно до рис. 2.

Отже, головна кнопкова форма створена, вона містить кнопку, після клацання на якій відкривається форма для реєстрації абітурієнтів.

Є недоліки. Інтерфейс головної форми не відповідає завданню. Для заміни стандартного вигляду форми переходять в режим конструктора. У вказаному режимі використовують вікно властивостей і змінюють вигляд форми відповідно до умови завдання.

Щоб головна кнопкова форма запускала автоматично після відкриття БД, звертаються до вікна **Параметрів: Файл/Параметры**. Надалі відкривають вкладнику **Текущая база данных** і у списку **Форма просмотра** обирають головну кнопкову форму, яка за умовою має назву **Вхід**.

Для автоматичного переходу до головної форми **Вхід** після закриття конкретної форми здійснюють редагування вкладки головної форми, яка посилається на конкретну форму: відповідній кнопці надають дію закриття поточної форми і повернення до форми **Вхід**.

Аналогічно здійснюють переходи між формами, формою і запитом, які мають бути в БД за умовою. Зупинимось на деяких особливостях.

Перехід до вікна **Подані заяви** — це перехід до результату відпрацювання запиту. Але запити, таблиці не є елементами головної кнопкової форми. Тому на головній формі створюють кнопки виклику макроса, а макрос створюється попередньо і є командою виклику, наприклад, запиту. Можна уникнути створення макросів. Створюється форма на основі запиту, а вже з головної форми викликається створена форма.

Для звернення користувача до конкретного запису, який відповідає вже зареєстрованому абітурієнту, необхідно створити параметричний запит, у якому до поля **Пароль** додає умова **[Ввести пароль]**. Якщо зна-

чення введенного паролю буде відповідати значенню однойменного поля, відкриється вікно **Подані заяви** (необхідно попередньо створити запит за умовою завдання). Якщо необхідно створити складений умовний вираз, у запиті створюють нове поле за допомогою будівника виразів з використанням стандартних функцій і логічних операцій.

Якщо спочатку була створена головна форма, а необхідно додати вкладку, то для редагування кнопкової форми звертаються до відповідних команд зміни і додавання елементів форми. Для виведення вікон з повідомленнями про помилки введення або перегляду даних користуються властивістю **Сообщение об ошибке** у вікні властивостей для поля. Але попередньо мають бути вказані умови на значення, а вже у разі невиконання цих умов і буде виведення повідомлення про помилку.

ЗАВДАННЯ 2. ТАБЛИЧНИЙ ПРОЦЕСОР

Проаналізуємо завдання. З умови видно, що будуть опрацьовані певним чином зведені дані, тобто на основі початкових даних таблиць необхідно створити нові таблиці **Вступники** і **Зведені дані**, для візуалізації перехресних даних останньої необхідно створити діаграми. Отже, у загальному вигляді опишемо вказівки виконання завдання.

1. Підготувати дані умови для використання до них команд створення зведених даних.

2. Створити зведені таблиці, діаграми.

3. Додати до певних аркушів кнопки керування за умовою.

5. Навести порядок у книзі, наприклад, приховати деякі стовпці, аркуші тощо, як того вимагається в умові завдання.

Результатом більш детального розгляду вказівок може бути зміна порядку їх виконання. Розглянемо, як виконати кожну вказівку.

Підготовка даних умови

На першому кроці роботи необхідно змінити формат подання початкових даних так, щоб можна було скористатись командою **Зведені таблиці** (або діаграми) стріч-

ки **Вставка** й отримати бажаний результат.

Отже, у таблиці, на основі якої буде зведення даних, мають бути стовпці (поля) таблиць, наданих умовою, з назвами **ЗНО** (бал ЗНО), **«С»** (середній бал атестату) та нові: **Факультет** і **Спеціальність**, оскільки саме за кількістю значень у цих полях створюється одна із зведених таблиць. У нові поля необхідно ввести значення, наприклад, у полі **Факультет** — значення **«Факультет кібернетики»**, а в **Спеціальності** — **«Програмна інженерія»**. Має сенс звести всі початкові дані в нову таблицю на окремому аркуші, а вже на основі отриманої таблиці здійснювати зведення даних. Є кілька варіантів виконання поставленої задачі.

Перший з них — просто копіювати дані з аркушів для кожної спеціальності в таблицю на окремому аркуші. До таблиці необхідно додати поля з назвами факультету і спеціальності, у першому рядку першої скопійованої таблиці ввести конкретні значення (фах) і скопіювати «протяжкою миші» на всі заповнені рядки конкретного фаху; надалі додати дані наступного аркушу і заповнити поля фаху. І так стільки разів, скільки в умові вказано спеціальностей. Але описаний варіант виконання копіювання спрацює для відомих кількостей заяв на кожну спеціальність, тобто, для фіксованої кількості вихідних даних кожної таблиці. Якщо в деяких таблицях знищити або додати дані про вступника, у скопійованих даних зміни не будуть відображені і виконання завдання не буде оцінено на всю кількість балів.

Динамічне використання вихідних даних реалізується в Excel функцією **СМЕШ**, наприклад, для опрацювання масивів, у даному випадку, двовимірних. Надалі, діапазон, масив і область будемо використовувати як синоніми. Функція **СМЕШ** дозволяє використати в подальшій роботі копію вихідних значень області посиланням на опорну комірку та кількості рядків, кількості стовпців, завдяки яким і визначається розмір області. Дана функція має ще два обов'язкових параметри, які дозволяють враховувати відстань (у рядках і стовпцях) необхідної прямокутної області від опорної комірки.

Формат функції такий **СМЕШ (АП, Р1, С1, Р2, С2)**, де АП — адресне

Изменение элемента кнопочной формы	
Текст:	Зарегистрироваться
Команда:	Открыть форму для добавления
Форма:	Абитуриенты
<div>ОК</div> <div>Отмена</div>	

Рис. 2. Створення підлеглих сторінок головної форми

посилання (зазвичай, абсолютне) на опорну комірку, **P1** і **C1** — відстань у рядках і стовпцях від опорної комірки до області вихідних даних; **C2** і **P2** — розмір у рядках і стовпцях області вихідних даних. Наприклад **СМЕШ (АП, 0, 0, 3, 4)** вказує на діапазон в 3 рядки, 4 стовпці, які рахуються одразу від адреси опорної комірки; а **СМЕЩ (АП, 0, 1, 0, 5)** вказує на 5 комірок того ж рядка, у якому знаходиться опорна комірка, але рахування здійснюється одразу з наступної за опорною коміркою. Завдяки такій властивості функції в цьому завданні можна динамічно використовувати діапазони вихідних даних — особливо це стосується фрази в завданні «Система повинна коректно працювати за будь-якого заповнення шести аркушів відповідно до умови».

Описана функція використовується в інших функціях для посилання на деяку область. Наприклад, функція **ИНДЕКС** повертає значення комірки, аргументами функції є масив (вказується діапазон комірок), номер рядка і стовпця, на перетині яких і знаходиться необхідна комірка. А оскільки діапазон комірок має бути динамічним, то й використовується функція **СМЕЩ**: **ИНДЕКС (СМЕЩ (АП;0;0;P1;C1);P2;C2)**, у якій:

АП — перша комірка з прізвищем абітурієнта певного фаху (має роль опорної);

0; 0 — саме від цієї комірки рахується діапазон даних, без відступів по рядках і стовпцях;

P1 — кількість абітурієнтів у таблиці певного фаху, яку необхідно обчислювати для врахування змінної кількості абітурієнтів: може бути як результатом обчислення певної формули, так і посиланням на комірку, яка утримує цю формулу;

C1 — кількість стовпців, які необхідно перенести в нову таблицю;

P2, C2 — номери рядків і стовпців вказаного діапазону: Якщо беруться прізвища, то **P2=1** і збільшується в процесі копіювання функції **ИНДЕКС(...)** у наступні комірки, а **C2=1**, бо в 1-му стовпці діапазону вказані прізвища, або **C2=4**, якщо мова йде про ЗНО.

Наведемо вигляд формули з конкретними значеннями, яка розміщена в комірці **C3** нового аркушу (дамо йому назву **ДодАркуш**):

**=ИНДЕКС (СМЕЩ (Аркуш1!\$C\$3;0;0;
Аркуш1!\$B\$3;4); ДодАркуш!B3;C\$1)**

У формулі в функції **СМЕЩ Аркуш1** означає ім'я аркуша зі списком абітурієнтів певного фаху. Вважаємо, що кількість заповнених рядків вказана в комірці **B3** того ж аркуша. Значення цієї комірки — кількість абітурієнтів таблиці з вказаного аркуша, яка обчислюється як результат певної функції (розглянуто нижче), а кількість стовпців є статичною і дорівнює 4, тобто це стовпці, у яких дані про ПІБ абітурієнта, суму набраних балів, середній бал атестату і бали за ЗНО.

Функція **ИНДЕКС** буде вставлена в комірку **C3** на додатковому аркуші (**ДодАркуш**). На цьому аркуші перший рядок, починаючи зі стовпця **C**, заповнимо числами 1, 2, 3, 4 тощо — так визначимо номер стовпця, а стовпчик **B**, починаючи з комірки **B3**, заповнимо числами арифметичної прогресії з кроком 1 — вони визначають номер рядка. Отже, якщо комірка **C3** знаходиться на перетині рядка 1 (значення **B3**) і стовпця 1 (значення **C1**) визначеного діапазону у функції **ИНДЕКС**, то функція поверне ПІБ 1-го абітурієнта, а ця ж функція, скопійована в комірку **C4**, поверне ПІБ другого абітурієнта, адже, у функції **ИНДЕКС** номер рядка визначено відносним адресним посилан

ням на комірку **B3**, яке відслідковує перехід до номера іншого рядка (результатом копіювання функції з **C3** в **C4** буде зміна в стовпці **B** на комірку **B4**, у якій значення 2). Так здійснюється вставка даних однієї таблиці на додатковий аркуш.

Здійсимо підрахунок вступників у кожній таблиці, тобто кількість заповнених (не пустих) комірок в окремому стовпці, наприклад, який містить поле ПІБ (кількість заповнених рядковими змінними комірок у вказаному діапазоні є результатом обчислення функції **СЧЕТЗ**, а результатом обчислення функції **СЧЕТ** є підрахування комірок із числами у вказаному діапазоні). Вступників буде на 1 менше, бо комірка із заголовком не є пустою коміркою. В окремій комірці, у формулі **B3**, необхідно здійснити вставку формули **СЧЕТЗ (C:C)–1**.

Порахувати кількість заповнених рядків у таблиці можна і за допомогою функції **ПОИСКПОЗ**, аргументами якої є значення елемента й масив, який утримує цей елемент, а повертає функція відносну позицію розташування елемента в масиві. Якщо елементом є неіснуюче значення, то функція поверне всю кількість елементів у масиві, що й є розв'язанням проблеми підрахунку заповнених рядків. Значенням елемента нехай буде рядкове значення "аяяяя": у стовпці ПІБ такого прізвища явно не існує. Масивом значень необхідно взяти весь стовпець — у кожній таблиці даних абітурієнтів це стовпчик **C**. Отже, формула використання функції така: **=ПОИСКПОЗ("аяяяя";C:C)**, а вставимо її, наприклад, у комірку **B2**. Функція поверне кількість заповнених комірок у вказаному стовпці. Кількість абітурієнтів буде на 2 меншою, оскільки пошук значення відбувається в даному випадку з першого рядка аркуша, а в таблицях перша комірка **C1** пуста, а наступна містить заголовок ПІБ (у комірці **B3** можна розмістити формулу **=B2–2**). Зазвичай, функція містить ще один аргумент, який визначає формат пошуку, він називається тип зіставлення. Для даної задачі значення типу має бути 1, тоді за правилами роботи функції вона поверне відносну позицію розташування найменшого значення в діапазоні (таке найменше значення є "", адже за прізвищем останнього абітурієнта міститься пуста комірка). Оскільки за замовчуванням тип зіставлення дорівнює 1, його можна не вказувати.

Отже, враховуючи вищезазначене, необхідно вставити формулу з функцією **ИНДЕКС(...)** у комірку **C3** додаткового аркуша, тоді скопіювати її «протяжкою миші» на наступні комірки рядка, додати назву факультету, у наступну комірку — назву спеціальності — перший рядок нової таблиці буде заповнено. Його необхідно виділити і скопіювати в рядки нижче. Вставлено значень буде стільки, скільки є абітурієнтів у таблиці **Аркуш1**. Останні рядки на додатковому аркуші будуть містити помилку — ті рядки, які перебільшують кількість розглядуваної висоти вхідного масиву даних (рядки, значення яких більші за кількість абітурієнтів). Таким чином здійснюється динамічне копіювання: стільки заповнено рядків вхідної таблиці, скільки й буде в іншому аркуші.

Але так, як описано вище, здійснюється вставлення даних з однієї таблиці, а необхідно мати дані усіх таблиць. Здійсимо деякі зміни в описаних формулах для динамічного відображення на додатковому аркуші даних усіх таблиць.

На додатковому аркуші автоматично заповнимо комірки стовпця **A** числами арифметичної прогресії з кроком **1** (необхідно активізувати комірку **A1**, використати стрічку **ГЛАВНАЯ**, інструмент **Заполнить**, вибрати **ВНИЗ** і ввести необхідні значення у вікні (прогресія арифметична, початкове значення **1**, крок **1**, заповнення по стовпцях, кінцеве значення **3000**).

В окремих комірках, наприклад **K2**, **K3** і нижче введемо посилання на комірки із значеннями, які відповідають кількості абітурієнтів кожного фаху. А в наступному стовпці суму абітурієнтів попередньо розглядуваних таблиць: в **L2** запишемо формулу **=L1+K2**, яку скопіюємо нижче, як показано в таблиці 1. Як зазначено вище, у комірках **В3** кожного аркушу визначено кількість заповнених рядків таблиці.

Таблиця 1

Допоміжна таблиця обчислення кількостей абітурієнтів

		Назва рядків таблиці	
Номер стовпців таблиці		K	L
	1		
	2	=Програмна інженерія_ФК!\$B\$3	=L1+K2
	3	=Програмна інженерія_ФІТ! \$B\$3	=L2+K3
	4	...	=L3+K4

Числа стовпця **A**, доки вони менші за значення **K2**, визначають рядок у функції **ИНДЕКС** для першої таблиці (аркуш **Програмна інженерія_ФК!**), для наступної таблиці, щоб почати з 1-го рядка діапазону в процесі копіювання «протяжкою» рядок буде визначено як значення стовпця **A** мінус значення **L2**, тощо. Тобто у формулі необхідно здійснити вкладеність умовних структур стільки разів, скільки розглядається таблиць з абітурієнтами. Наведемо фрагмент формули комірки **C3** допоміжного аркуша:

```
=ЕСЛИ(A1<=$L$2;  
ИНДЕКС (СМЕЩ (ПФК!$C$3;0;0;$K$2;4);$A1;C$1);  
ЕСЛИ(A1<=$L$3;  
ИНДЕКС (СМЕЩ (ПФІТ!$C$3;0;0;$K$3;4);  
$A1-$L$2;C$1);ЕСЛИ...))...
```

У формулі скорочено представлено ім'я аркушів: **ПФК** означає **Програмна Інженерія Факультет Кібернетики**, а **ПФІТ** — відповідно **Програмна Інженерія Факультет Інформаційних Технологій** тощо.

Щоб підсумкова таблиця мала дані про спеціальність і факультет, користуються умовною функцією з використанням тієї ж таблиці 1.

Отже, усе вищезазначене дозволяє отримати дані абітурієнтів на всі спеціальності факультетів на одному аркуші. Є невеличка примітка: якщо з якоїсь таблиці знищити рядок з даними на абітурієнта, останні рядки загальної таблиці будуть утримувати текст, який вказує на помилкове посилання. Щоб усе було красиво, можна застосувати функцію **ЕСЛИОШИБКА**. Її аргументом буде вигляд вищезазначеної формули; функція логічного типу повертає значення аргументу, якщо помилки відсутні й будь-яке, вказане користувачем, наприклад **0**, якщо неправильне посилання. Тоді останні рядки будуть утримувати **0**.

Створення зведених таблиць

Зведена таблиця, яку необхідно створити, можна назвати за аналогією з типом звіту в системі керування базами даних перехресною таблицею. Дійсно, заголовками рядків будуть назви факультетів, стовпців (полів) — назви спеціальностей, а комірки таблиці визначають

фах, адже кожна комірка — це перетин рядка (відповідного факультету) і стовпця (спеціальності): у комірках таблиці кількість абітурієнтів певного фаху.

Створення зведеної таблиці здійснюється за стандартним алгоритмом використання команди **Сводная таблица** стрічки **ВСТАВКА**. Заголовки рядків — поле **Факультет**; заголовки стовпців — поле **Спеціальність**; значення — **кількість ПБ**, тобто кількість абітурієнтів, яких можна порахувати за значеннями в полі **ПБ** для даного фаху. Таблицю необхідно відредувати на відповідність її зовнішнього вигляду зразковому із завдання. Візуалізація даних про процес вступу — у діаграмі: після виділення окремих областей зведеної таблиці необхідно здійснити вставку кругової діаграми. Область діаграми має сенс доповнити написом про спеціальності, як це показано на зразку.

Таблиця аркушу **Вступники** створена як зведена на основі поля додаткового аркуша **ПБ**, яке у зведеній таблиці є заголовком рядків, і кількість однакових значень цього ж поля є значенням зведеної таблиці. Репшта необхідних полів, вказана в умові завдання (**Бал ЗНО і Бал аттестату**), додається як результат функції **ВПР** (значення поля **ПБ** зведеної таблиці; **Таблиця** з додаткового аркуша; номер необхідного стовпця). Щоб врахувати всі заповнені рядки таблиці з додаткового аркуша, замість таблиці можна вставити назви стовпців. 3-й стовець від стовпця **ПБ** містить **Бал аттестату**, а в 4-му стовпці значення **Балу ЗНО**. Отже, перепишемо формулу для визначення балу ЗНО в комірку **E3** (2-й рядок таблиці аркуша **Вступники** є рядком заголовку, поле **ПБ** — це стовець **C**, **Поле Бал ЗНО** — стовець **E**, **Поле Бал аттестату** — стовець **F**):

=ВПР(\$C3;ДодАркуш!\$C:\$F;4).

Аналогічно визначається бал аттестату й копіюванням «протяжкою миші» заповнюємо всю таблицю. Налагоджуємо вигляд таблиці відповідно до вимог завдання.

Додавання кнопок керування

Розглянемо, як вставити кнопки керування і закріпити за ними оновлення створених таблиць для відслідковування змін у вихідних таблицях. Як кнопки керування можна вставити малюнок із файлу; малюнок, створений засобами Microsoft Office; кнопку керування зі стрічки **РАЗРАБОТЧИК**. У контекстному меню вставленого об'єкта необхідно вибрати команду **Назначить макрос**.

У цьому завданні результатом відпрацювання макросів є оновлення даних зведених таблиць. Отже, увімкнути запис макроса необхідно перед використанням вказаної команди. Команду зручно викликати з контекстного меню, у даному випадку, комірок із значеннями. Оскільки діаграма будується на значеннях таблиці, її створення не включають в команди макроса. Для коректної роботи за записом макросів необхідно здійснити налаштування табличного процесора:

- після запуску команди **ФАЙЛ** — **Параметры**, у вікні **Параметры Excel** виділяють пункт **Центр управління безпекою** й активізують кнопку **Параметры центра управления безопасностью**;
- у вікні **Центр управления безопасностью** виділяють пункт **Параметры макросов** й активізують команду **Включить все макросы**;
- необхідно перезапустити табличний процесор.

(Далі буде)