

**МІНІСТЕРСТВО КУЛЬТУРИ ТА
ІНФОРМАЦІЙНОЇ ПОЛІТИКИ УКРАЇНИ**

ХАРКІВСЬКА ДЕРЖАВНА АКАДЕМІЯ КУЛЬТУРИ

Факультет культурології та соціальних комунікацій

Кафедра цифрових комунікацій та інформаційних технологій

ТЕХНОЛОГІЯ ПРОГРАМУВАННЯ ІНТЕРФЕЙСІВ

Програма та навчально-методичні матеріали до курсу
для здобувачів першого (бакалаврського) рівня вищої освіти зі
спеціальності 029 «Інформаційна, бібліотечна та архівна справа»
ОП «Інформаційна та документаційна діяльність»

Харків – 2024

УДК 004.42(073)

Т 38

Рекомендовано до видання науково-методичною радою ХДАК
(протокол № 13 від 29.04.2024 р.)

Рекомендовано кафедрою цифрових комунікацій та
інформаційних технологій
(протокол №15 від 26.03.2024 р.)

Рецензенти:

Брусенцев В. О., доцент, кандидат технічних наук,
доцент кафедри цифрових комунікацій
та інформаційних технологій ХДАК

Білова Т. Г., доцент, кандидат технічних наук,
доцент кафедри системотехніки

Харківського національного університету радіоелектроніки

Укладач:

Побіженко І. О., доцент, кандидат технічних наук,
доцент кафедри цифрових комунікацій
та інформаційних технологій ХДАК

Технологія програмування інтерфейсів : програма та навч.-метод.
Т 38 матеріали до курсу для здобувачів першого (бакалавр.) рівня вищої
освіти зі спеціальності 029 «Інформаційна, бібліотечна та архівна
справа», ОП «Інформаційна та документаційна діяльність» / М-во
культури та інформ. політики України, Харків. держ. акад. культури, Ф-
т культурології та соц. комунікацій, Каф. цифрових комунікацій та
інформ. технологій ; [уклад. І О. Побіженко]. Харків : ХДАК, 2024. 57 с.

Навчальний курс «Технологія програмування інтерфейсів» є важливою складовою підготовки бакалаврів за спеціальністю 029 «Інформаційна, бібліотечна та архівна справа». Зміст курсу розкриває особливості однієї з суттєвих частин науки інформаційних технологій — програмування інструментальних засобів електронних документів. Основну увагу направлено на вивчення об'єктно-орієнтованих методів програмування на базі UNO-технології мовою ООО Basic.

УДК 004.42(073)

© Харківська державна академія культури, 2024 р.

© Побіженко І.О., 2024 р.

**ПРОГРАМА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ
ТЕХНОЛОГІЯ ПРОГРАМУВАННЯ ІНТЕРФЕЙСІВ**

Опис навчальної дисципліни

Найменування показників	Галузь знань, напрям підготовки, освітньо-кваліфікаційний рівень	Характеристика навчальної дисципліни	
		денна форма навчання	заочна форма навчання
Кількість кредитів – 3	Галузь знань 0201. Культура	Вибіркова	
	Напрямок підготовки 029 «Інформаційна бібліотечна та архівна справа»		
	Освітньо-професійна програма Інформаційно-документаційна діяльність		
3-й			
Загальна кількість годин - 90		Семестр	
		5-й	
Тижневих годин для денної форми навчання: аудиторних 2 самостійної роботи студента – 4	Ступінь: бакалавр	10 год.	год.
		Семінарські	
		год.	год.
		Практичні	
		20 год.	год.
		Самостійна робота	
		60 год.	

Примітка

Співвідношення кількості годин аудиторних занять до самостійної і індивідуальної роботи становить:

для денної форми навчання – 30/60

Мета та завдання навчальної дисципліни

Мета курсу – наданні студентам теоретичних та практичних знань з сучасних технологій поширення згідно з потребами користувача функціональних можливостей офісних додатків операційних систем, так і їх використання у сучасній бібліотечно-інформаційній діяльності.

Загальні та фахові компетентності, які формує дисципліна.

Загальні компетентності	ЗК1) здатність до абстрактного мислення, аналізу та синтезу;
	ЗК2) здатність генерувати нові ідеї (креативність);
	ЗК3) вміння виявляти, ставити та вирішувати проблеми;
	ЗК4) здатність мотивувати людей рухатися до спільної мети;
	ЗК5) здатність виявляти ініціативу та підприємливість;
	ЗК6) здатність оцінювати та забезпечувати якість виконуваних робіт.
Професійні компетентності	ПК1) здатність до формування ефективної системи управління інформаційною, бібліотечною та архівною діяльністю;
	ПК3) здатність застосовувати технології для вирішення практичних, управлінських, науково-дослідних і прогностичних завдань у професійній діяльності
	ПК6) здатність застосовувати комп'ютерні технології з метою вдосконалення професійної діяльності, розуміти принципи проектування та функціонування автоматизованих БД, семантичного вебу, веб-сервісів та соціальних медіа;
	ПК9) здатність організувати та реалізувати маркетингову діяльність інформаційних установ;
	ПК12) здатність здійснювати інформаційний моніторинг

Програмні результати навчання (відповідно до освітньо-професійної програми).

Програмні результати навчання	ПРН1. Знати і розуміти наукові засади організації, модернізації та впровадження новітніх технологій в інформаційній, бібліотечній та архівній діяльності.
	ПРН2. Впроваджувати та використовувати комунікаційні технології у соціальних системах, мультимедійне забезпечення інформаційної діяльності, технології веб-дизайну та веб-маркетингу.
	ПРН8. Використовувати знання технічних характеристик, конструкційних особливостей, призначення і правил експлуатації комп'ютерної техніки та офісного обладнання для вирішення технічних завдань спеціальності.
	ПРН9. Оцінювати можливості застосування новітніх інформаційно-комп'ютерних та комунікаційних технологій для вдосконалення практик виробництва інформаційних продуктів і послуг.
	ПРН10. Кваліфіковано використовувати типове комп'ютерне та офісне обладнання.
	ПРН11. Здійснювати пошук інформації в різних джерелах для розв'язання професійних завдань.
	ПРН12. Застосовувати сучасні методики і технології автоматизованого опрацювання інформації, формування та використання електронних інформаційних ресурсів та сервісів.

Засоби оцінювання та методи демонстрування результатів навчання та підсумкова форма контролю

Засобами оцінювання та демонстрування результатів навчання та підсумкова форма контролю є:

- залік, іспит;
- тести;
- виконання практичних робіт;

● презентації результатів виконаних студентами практичних завдань.

Критерії оцінювання результатів навчання за 100 бальною рейтинговою шкалою:

1. Високий рівень знання матеріалу курсу (90 – 100 балів, А, відмінно):

вільне володіння понятійним апаратом,

вільна орієнтація в матеріалі курсу і здатність відповісти на різні питання;

високий рівень виконання програмних завдань.

2. Достатньо високий рівень знання матеріалу курсу (74 – 89 балів, С, В, відмінно):

добре володіння основними термінами і поняттями,

досить вільна орієнтація в матеріалі курсу і здатність відповісти на основні питання;

добрий рівень виконання програмних завдань.

3. Посередній рівень знання матеріалу курсу (60 -73 балів, D, E, задовільно):

посередній рівень володіння основними термінами і поняттями,

недостатньо вільна орієнтація в матеріалі курсу і здатність відповісти на окремі питання;

посередній рівень виконання програмних завдань.

4. Незадовільний рівень з можливістю повторного складання (35 – 59 балів, FX, незадовільно):

незнання окремих термінів та понять,

незнання основного матеріалу курсу і неспроможність відповісти на більшість питань;

незадовільний рівень виконання окремих програмних завдань.

5. Незадовільний рівень з обов'язковим повторним вивченням дисципліни (0 – 34 бали, F, незадовільно):

незнання основних термінів та понять,

незнання матеріалу курсу і неспроможність відповісти на питання;

нездатність до виконання більшості програмних завдань.

Основним критерієм успішності вивчення здобувачем/здобувачкою освіти навчальної дисципліни є отримання ним/нею не менш як 60 балів зі 100 за сукупністю усіх видів робіт: складання підсумкового іспиту, участь у лекційних заняттях, відповіді на семінарських заняттях, виконання завдань для самостійної роботи.

Кількість тижнів для теоретичної підготовки, кількість розділів та тем: 15 тижнів, 2 розділи та 7 тем.

ЗМІСТ І СТРУКТУРА НАВЧАЛЬНОЇ ДИСЦИПЛІНИ

Структура навчальної дисципліни

Назви розділів	Кількість годин			
	Денна форма			
	Усього годин	у тому числі		
Л		Лаб	С. р.	
2	3	4	6	7
Розділ 1. Основи сучасної методології програмування інтерфейсів користувача.				
Тема 1. Об'єктно-орієнтоване програмування (ООП) — базова технологія сучасного програмного забезпечення засобів інтерфейсу користувача	10	2	2	6
Тема 2. Програмне забезпечення елементів управління у інтерфейсах користувача.	10	2	2	6
Разом за розділом 1	20	4	4	12
Розділ 2. Типові методики програмування інтерфейсів користувача.				

Тема 3. Загальні методики організації інтерфейсів на основі діалогу.	17	1	4	12
Тема 4. Типізація діалогів. Обчислювальні діалоги.	17	1	4	12
Тема 5. Довідкові діалоги.	10	2	2	6
Тема 6. Діалоги сортування та пошуку.	26	2	6	18
Разом за розділом 2	60	6	16	48
Усього годин	90	10	20	60

ЗМІСТ КУРСУ

«Технологія програмування інтерфейсів» як навчальна дисципліна. Теоретичне та практичне значення курсу. Предмет, завдання, зміст і структура курсу, його місце у системі професійної підготовки фахівця з спеціальності 029 Інформаційна, бібліотечна та архівна справа. Взаємозв'язок курсу з іншими дисциплінами.

Джерелознавча база вивчення навчальної дисципліни: основна і додаткова література.

РОЗДІЛ 1

ОСНОВИ СУЧАСНОЇ МЕТОДОЛОГІЇ ПРОГРАМУВАННЯ ІНТЕРФЕЙСІВ КОРИСТУВАЧА

Тема 1. Об'єктно-орієнтоване програмування (ООП) — базова технологія сучасного програмного забезпечення засобів інтерфейсу користувача.

Поняття інтерфейсу користувача та загальні вимоги до інтерфейсів. Елементи управління — базові програмні засоби програмування інтерфейсів користувача. Поняття класу, об'єкту, та членів класу. Поняття методу, властивості, події об'єкта. Класифікація основних елементів управління та їх властивостей.

Практичне заняття: Знайомство з елементами управління, як основними об'єктами створення інтерфейсів користувача.

Самостійна робота: Поняття інтерфейсу користувача і основні вимоги до нього.

Тема 2. Програмне забезпечення елементів управління у інтерфейсах користувача.

Поняття форми та діалогу. Оголошення об'єкту та синтаксичні засоби доступу до членів його класу. Технологія створення елементів управління на формах документів. Поняття програмної та візуальної моделей елемента управління. Програмні засоби доступу до моделей елементів управління на формах.

Практичне заняття: Програмні моделі елементів управління.

Самостійна робота: Основні зведення про проектування інтерфейсів користувача. Особливості проектування графічного інтерфейсу.

РОЗДІЛ 2

ТИПОВІ МЕТОДИКИ ПРОГРАМУВАННЯ ІНТЕРФЕЙСІВ КОРИСТУВАЧА

Тема 3. Загальні методики організації інтерфейсів на основі діалогу.

Організація діалогів засобами середовища OpenOffice.org (LibreOffice.org): основні маніпуляції з вікнами та командами створення діалогів. Програмні засоби завантаження моделі діалогу в оперативну пам'ять. Засоби розміщення елементів управління на діалозі.

Програмні засоби доступу до елементів управління на діалогах. Поняття багатосторінкового діалогу. Технологія розміщення елементів управління на багатосторінкових діалогах та управління сторінками.

Практичне заняття: Організація простого діалогу.

Практичне заняття: Багатосторінкові діалоги.

Самостійна робота: Елементи управління, як діалогові об'єкти. Вікна, піктограми та їх проектування.

Тема 4. Типізація діалогів. Обчислювальні діалоги.

Класифікація типових діалогів за призначенням: обчислювальні, командні, довідкові, сортувань та пошуку. Загальна структура обчислювального діалогу. Засоби обміну даними на рівні користувач — діалог. Особливості програмування елементів управління обчислювального діалогу.

Практичне заняття: Обчислювальні діалоги.

Самостійна робота: Обчислювальні діалоги. Типи та різновиди меню. Кнопки, перемикачі, прапорці.

Тема 5. Довідкові діалоги.

Поняття довідкового діалогу та його основні функції. Засоби зберігання даних та особливості доступу до даних.

Програмна організація алгоритмів вилучення даних для довідки. Інструкція Select Case та приклади її програмування

Практичне заняття: Довідкові діалоги.

Самостійна робота: Списки. Текстові поля.

Тема 6. Діалоги сортування та пошуку.

Алгоритми сортування: загальна характеристика. Основні різновиди алгоритмів сортування. Порівняння алгоритмів сортування по швидкодії. Алгоритм сортувань бульбашками та його програмна реалізація.

Алгоритми пошуку по ключу в неупорядкованому та упорядкованому масивах. Особливості пошуку даних у файлах. Програмні засоби відкриття та закриття файлів. Програмні засоби обміну даними на рівні файл — макрос. Діалоги створення файлів та діалоги пошуку і обчислень у файлах.

Практичне заняття: Діалоги сортувань.

Практичне заняття: Діалоги підкочувань файлів.

Практичне заняття: Діалоги пошуку у текстових файлах.

Самостійна робота: Засоби угруповання прокрутки та індикації. Проектування засобів підтримки користувача. Методика проектування інтерфейсів: від загального до окремого.

МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ДО ПРАКТИЧНИХ ЗАНЯТЬ І САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ

Теми практичних занять

№ з/п	Назва теми	Кількість годин
1	Знайомство з елементами управління, як основними об'єктами створення інтерфейсів користувача.	2
2	Програмні моделі елементів управління.	2
3	Організація простого діалогу.	2
4	Багатосторінкові діалоги.	2

5	Обчислювальні діалоги.	4
6	Довідкові діалоги.	2
7	Діалоги сортувань.	2
8	Діалоги підкочувань файлів.	2
9	Діалоги пошуку у текстових файлах.	2
	Разом	20

Теми самостійних робіт

№ з/п	Назва теми	Кількість годин
1	Поняття інтерфейсу користувача і основні вимоги до нього.	6
2	Основні зведення про проектування інтерфейсів користувача. Особливості проектування графічного інтерфейсу.	6
3	Елементи управління, як діалогові об'єкти. Вікна, піктограми та їх проектування	12
4	Типи та різновиди меню. Кнопки, перемикачі, прапорці..	12
5	Списки. Текстові поля.	6
6	Засоби угруповання прокрутки та індикації. Проектування засобів підтримки користувача. Методика проектування інтерфейсів: від загального до окремого.	18
	Разом	60

**Завдання до практичних занять та самостійної роботи
здобувачів**

Теми практичних занять

Практичне заняття № 1.

**Тема: Знайомство з елементами управління, як основними
об'єктами створення інтерфейсів користувача.**

Мета: Ознайомитися з моделлю побудови елементів управління, як основних об'єктів створення інтерфейсів користувача.

Завдання:

Опанувати методику створювати елементи управління, як основними об'єктами створення інтерфейсів користувача.

Зміст та послідовність виконання заняття.

Завдання 1. У середовищі операційної системи створити особисту папку, що позначена як «ПБ студента_Програмування інтерфейсів_Звіти». У цій папці послідовно щодо виконання занять повинно бути створено папки з назвами «Практичне заняття №1», «Практичне заняття №2» і далі, де відобразатиметься виконання звітів практичних занять по данному курсу. На підставі цих особисто виконаних та зарахованих робіт студенту будуть нараховуватись бали викладачем.

Завдання 2. У середовищі Open Office.org або (Libre Office.org) створіть новий текстовий документ, де засобами меню Вид виведіть на екран панель елементів управління. Підведіть мишку до кожного з елементів управління і зверніть увагу на те, який тип має кожен з елементів управління. Визначить у звіті скільки всього та яких елементів управління може бути задане користувачем для проектування інтерфейсів засобами даної панелі.

Завдання 3. Засобами мишки розташуйте у площині документа елемент управління Кнопку. Засобами контекстного меню (Елемент управління) роздрукуйте на екрані властивості даного елемента

управління. Уважно розгляньте та відобразіть у звіті призначення кожної з властивостей. Такі призначення робляться розробниками програмного забезпечення на підставі «інтуїтивно прозорого інтерфейсу». Однак, якщо деякі властивості можуть визивати сумніви щодо їх тлумачення, зверніться до довідки або викладача.

Завдання 4. Аналогічно до попереднього завдання розгляньте властивості решти елементів управління та відобразіть їх призначення у звіті. Але для кожного наступного елемента відображайте тільки ту інформацію, що не визначено для попередніх.

Завдання 5. Здобудьте навички програмування елементів управління у режимі розробки. Для цього скористатись контекстним меню (див. малюнок зліва) одного з них, наприклад, Кнопки. Такі програмування можна виконувати, змінюючи безпосередньо властивості обраного елемента, або скористатись засобами вікон, що підтримуються контекстним меню. У звіті відобразити, як програмувати розміри, кут, ім'я та підписи даного елемента за допомогою вікна Положення та розмір.

Завдання 6. Засобами відображення властивостей елемента управління у режимі розробки запрограмуйте кнопку напис різними шрифтами, фонами або малюнками. У звіті відобразити по одному з прикладів кожного з таких властивостей.

Створіть звіт з результатів виконання практичного заняття.

Практичне заняття № 2.

Тема: Програмні моделі елементів управління**Мета:** засвоїти програмні моделі елементів управління**Завдання:**

Опанувати методику роботи з програмними моделями елементів управління

Зміст та послідовність виконання заняття:

Завдання 1. У середовищі Open Office.org або (Libre Office.org) створіть новий текстовий документ, де засобами меню *Вид* виведіть на екран панель елементів управління. Розташуйте на екрані елемент управління *Кнопка* і надайте йому ім'я **Кнопка 1**. Суть наступних завдань завдання буде полягати у тому, щоб дістатися моделі будь якого з елементів управління програмним шляхом.

Завдання 2. Спочатку дістанемось нашої *Кнопки*. Модель кожного з елементів управління має доступ через метод `GetByName` об'єкта `Forms`. Тому спочатку створіть макрос у бібліотеці `Standard` того файлу, що було відкрито. Наприклад, так:

Sub Main**Dim Doc As Object****Dim Form As Object****Dim Ctl As Object****Doc = ThisComponent****Form = Doc.DrawPage.Forms.GetByIndex(0)****Ctl = Form.getByName("Кнопка 1")**

Msgbox(Ctl.name)

End Sub

Завантажте цей макрос на виконання і упевніться у тому, що модель **Ctl** дійсно підтримує елемент управління *Кнопка*, що має ім'я **Кнопка 1**. Як що усе зроблено вірно, то у вікні повідомлень буде відповідно визначено: **Кнопка 1**. У протилежному випадку налагодьте макрос.

Завдання 3. Уважно розгляньте визначення властивостей моделі *Кнопка*. і, на основі цього опису встановіть програмним шляхом властивості кнопки:

- колір фону: синій;
- шрифт: Times New Roman;
- розмір шрифту: 14.
- надпис: ОК.

Наприклад, для того щоб змінити колір кнопки на колір, що визначено користувачем, можна записати такий фрагмент програми:

Sub Main

Dim Doc As Object

Dim Form As Object

Dim Ctl As Object

Doc = ThisComponent

Form = Doc.DrawPage.Forms.GetByIndex(0)

Ctl = Form.getByname("Кнопка 1")

Msgbox(Ctl.name)

Ctl.BackgroundColor = &Hffffaa

End Sub

У даному випадку маємо приємний жовтий колір на кнопці. При завданні кольору доцільно визначати його символами шістнадцятиричної системи. У даному випадку перші два з символів **&H** визначають, що подальші цифри будуть шістнадцятиричні: по дві у кожному з байтів. Так у прикладі маємо: **ff, ff, aa**, що надає: 255 червоного, 255 – зеленого та 170 – рівнів синього відтінків.

Завдання 4. Це завдання виконайте аналогічно попередньому, але для документу Calc. При його виконанні майте на увазі, що при роботі з електронними таблицями необхідна проміжна стадія: визначитись не тільки з документом, але й з листом, на якому маємо елемент управління. Тому після того, як визначено **Doc = ThisComponent**, слід створити об'єкт, що підтримує відповідний аркуш:

Dim Sheet As Object

Sheet = Doc.Sheets.GetByIndex(0)

До об'єкта Sheet слід буде і звертатися при виконанні завдання зміни кольору кнопки.

При виконанні завдання не обов'язково заводити кнопку. Рекомендується користуватися й іншими елементами управління, наприклад, перемикачем або прапорцем.

Завдання 5. Перегляньте всі списки властивості елементів управління [3, с. 179 -183] і зверніть увагу на те, що там відсутні дані про геометричні параметри цих елементів. Поясніть собі, що справа в тому, що геометрична модель елемента управління розглядається як малюнок на макеті форми. Тому до нього треба мати окреме звертання. Знайдіть об'єкт-малюнок, що відповідає моделі елемента управління, якого завдано унікальним іменем. Зробіть це так.

Перелічіть у моделі форми усі малюнки що є, фільтруйте з них ті, що належать до малюнків елементів управління, а з останніх виберіть єдину – ту, що має завдане ім'я. Потім виконайте налаштування геометричних розмірів елемента управління. У прикладі застосовано елемент управління текстове поле, що має ім'я TextBox1. Його ширина та висота повинні скласти по 5000 пунктів.

Приклад відповідного фрагменту програми для кнопки наведено нижче.

Sub Main

Dim Doc As Object

Doc = ThisComponent

Dim I As Integer

'починаємо пошук малюнка з заданим ім'ям TextBox1

For I= 0 to Doc.DrawPage.Count -1

Shape = Doc.DrawPage(I)

**If HasUnoInterfaces(Shape,
"com.sun.star.drawing.XControlShape") Then**

If Shape.Control.Name = "TextBox1" Then

Exit for

End If

End If

next I

'змінюємо розміри елемента управління

```
dim oSize as object
```

```
oSize = Shape.getSize()
```

```
oSize.width = 5000
```

```
oSize.Height = 5000
```

```
Shape.setsize(oSize)
```

```
end sub
```

Цю програму налагодьте та випробуйте для кількох елементів управління.

Завдання 6. Виконайте завдання, аналогічне попередньому, але як геометричні параметри завдайте координати центру елемента управління. Так у фрагменті програми, що надано нижче, координати центру змінюються по горизонталі на 150 відсотків, а по вертикалі на 200 відсотків.

```
dim oPosition as object
```

```
oPosition=Shape.getPosition()
```

```
oPosition.x= oPosition.x*1.5
```

```
oPosition.y= oPosition.y*2
```

```
Shape.setPosition(oPosition)
```

Для налагодження програми цей фрагмент можна вставити у попередній файл безпосередньо перед **end sub**.

Створіть звіт з результатів виконання практичного заняття.

Практичне заняття № 3.

Тема: Організація простого діалогу

Мета: опанувати загальну технологію створення діалогових інтерфейсів.

Зміст та послідовність виконання заняття:

Завдання 1. У середовищі Open Office.org або (Libre Office.org) створіть новий текстовий документ, де засобами меню *Вид* виведіть на екран панель елементів управління.

- Засобами меню *Сервіс - Макроси - Управління макросами - OpenOffice.org Basic - кнопка Управління* створіть за допомогою вкладки *Модуль* новий модуль у середовищі відкритого файлу. Надайте йому ім'я за бажанням. У цьому модулі будемо створювати макроси, що підтримують майбутній діалог.

- Засобами вкладки *Діалог* створіть новий діалог.

- Введіть ім'я діалогового вікна і натисніть кнопку *Редагувати*. буде відкрито вікно BASIC, в якому міститься порожнє діалогове вікно.

- Якщо *Панель елементів управління* не відображається, клацніть стрілку поряд із знаком *Вставити елементи управління*, аби відкрити *Панель елементів управління*. У результаті виконаних дій буде створено пусте вікно діалогу.

Завдання 2. Скористайтеся інструментами на *Панелі елементів управління* у редакторі діалогових вікон щоб створити діалогове вікно і додати наступні елементи управління: *Прапорець* з ім'ям "Checkbox1", *Підпис* з ім'ям "Label1", *Кнопку* з ім'ям "Commandbutton1" і *Список* з ім'ям "Listbox1". Визначення усіх імен перевірте або задайте через списки властивості кожного з елементів управління.

Завдання 3. Відобразіть діалогове вікно у документі. Для цього скористайтеся типовим набором інструкцій у макросі : **StartDialog1:**

Dim oNameDialog AS Object

Sub NameDialogShow

DialogLibraries.LoadLibrary("Standard")

```
oNameDialog = CreateUnoDialog( DialogLibraries.Standard .
NameDialog )
```

```
oNameDialog.Execute()
```

End Sub

При користуванні цими інструкціями слід навчитися відрізняти імена об'єктів власної програми від імен об'єктів у середовищі OOo. Так у фрагменту вище **NameDialog** визначає те ім'я, що надано діалогу при його створенні. Усі імена, що починаються на **o** розглядаються як імена об'єктів власної програми. Налаштуйте цей макрос згідно з вказаними вище пропозиціями і помістіть у модуль, що створено при виконанні **Завдання1**. Завантажте макрос **NameDialogShow** на виконання. Якщо усе зроблено вірно, то у документі повинен з'явитися діалог.

Завдання 4. Виконайте налаштування кожного з елементів управління з тим, щоб він міг виконувати відповідну йому роботу. Але спочатку клацніть мишкою у робочому полі кожного з елементів управління і спостерігайте реакцію на відповідь. У даному випадку буде змінюватись тільки зовнішній вигляд прапорця та кнопки. Так і повинно бути. Щоб виконати налаштування елемента управління треба написати для нього макроси згідно з алгоритмом виконання цієї роботи.

Хай прапорець при установці його у стан «так» змінює напис на *Підпису* «Кнопку активовано» і при цьому кнопка стає дійсно активною. У протилежному разі хай буде напис «Кнопка пасивна» і її заблоковано.

Нижче дано фрагмент програми, що може виконувати такі дії. Але не рекомендується цей фрагмент розміщувати у модулі простим копіюванням. Щоб програма працювала в цілому, слід добре розібратися з іменами об'єктів, які взаємодіють у всьому модулі і налаштувати цю відповідність у окремих макросах.

Приклад виконання вправи для об'єкту **oDialog1** дано на фрагменту нижче:

Sub ActiveButton

```
oCheckBox1 = oDialog1.GetControl("CheckBox1")
oCommandButton1 = oDialog1.GetControl("CommandButton1")
oLabel1 = oDialog1.GetControl("Label1")
    if oCheckBox1.model.state then
        oLabel1.model.Label="Кнопку активовано"
        oCommandButton1.model.Enabled=True
    else
        oLabel1.model.Label="Кнопка пасивна"
        oCommandButton1.model.Enabled=False
    end if
end Sub
```

При створенні подібних програм спочатку треба дістатися до моделі кожного з елементів управління, що задіяні у алгоритмі. Ці функції виконуються у перших трьох рядках процедури **ActiveButton**. Далі, застосуванням інструкції **if**, аналізується статус прапорця і виконуються відповідні дії. Розташуйте макрос у тому ж самому модулі, де й створено макрос завантаження діалогу.

Після написання подібного макросу треба зв'язати макрос з подією, яка відбувається при натисканні мишкою елемента управління. Для цього поверніться до діалогу, розкрийте властивості прапорця і перейдіть на вкладку подій. У даному випадку слід зв'язати подію змінення стану прапорця з макросом **ActiveButton**. Для цього треба скористатися кнопкою, що визначено багатьма крапками у вікні подій. Решта дій здається прозорою.

Завантажте макрос запуску діалогу на виконання. Якщо все зроблено вірно, то запрограмовані дії повинні відобразитись у діалогу.

Завдання 5. Створіть макрос, що виконує такі дії: при натисканні на активну кнопку створює у *Списку* список імен всіх елементів управління, що маємо у діалогу, а потім дезактивує кнопку.

Завдання 6. Створіть макрос, що при подвійному щиглику на *Списку* видаляє всі дані зі списку.

Створіть звіт з результатів виконання практичного заняття

Практичне заняття № 4.

Тема: Багатосторінкові діалоги.

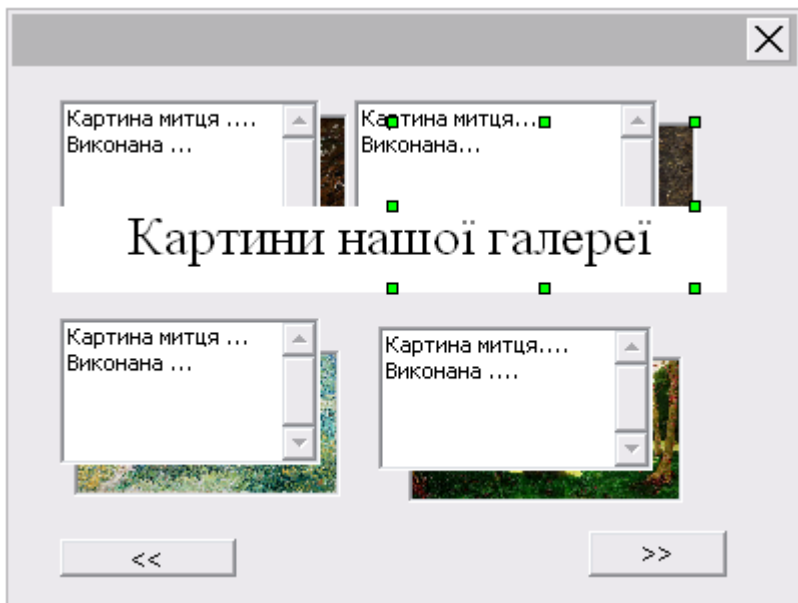
Мета: опанувати технологію створення багатосторінкових діалогів.

Завдання. Створити та випробувати багатосторінкові діалоги.

Зміст та послідовність виконання заняття:

Завдання 1. У середовищі текстового редактора створити пустий діалог. Цей діалог буде відображати елементи управління в залежності від того, яку сторінку обрано для перегляду діалогу. Назвіть цей діалог за бажанням.

Головне завдання буде таке. На першій сторінці хай будуть представлені напис «Картини нашої галереї» та кнопки для переходу на наступну та попередню сторінки (з відповідним блокуванням по межах переходів). На другій сторінці — буде видно чотири графічних зображення, наприклад, картини митців, а також дві кнопки для переходів на наступну сторінку або попередню сторінку. На третій сторінці — буде представлено тексти, що надають коротку інформацію про кожну з картин з другої сторінки, а також обидві кнопки для переходу на сторінки.



Розміщення елементів управління виконайте так, щоб не звертати особливої уваги на то, як розміщено елементи на сторінках. Наприклад, на даному діалозу у першу чергу можна розмістити головний матеріал — картини у графічних елементах управління, зверху них розмістити текстові поля, а потім поле напису. Може вийти так, як дано на рисунку зліва.

Завдання 2. Точно визначте, які елементи управління будуть на якій із сторінок представлено. Рахуйте сторінки, починаючи з одиниці. Далі задайте усім елементам управління та самі ж значення у властивостях Сторінка (Крок). Оскільки не можна визначити список сторінок у даній властивості, то задайте для обох кнопок значення нуль. Тоді вони можуть бути видимі на усіх сторінках. Там же, де вони не потрібні, визначте засоби програмування меж на відповідному етапі.

Завдання 3. Створіть новий модуль з макросом, що дано на примірнику нижче та випробуйте його.

Dim oDialog2 AS Object

Sub DialogShow

DialogLibraries.LoadLibrary("Standard")

oDialog2 =

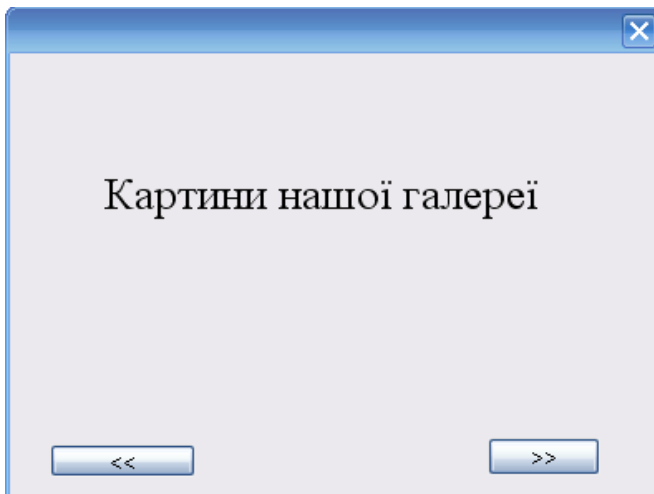
CreateUnoDialog(DialogLibraries.Standard.Dialog2)

oDialog2.Model.Step=1

oDialog2.Execute()

End Sub

Зверніть увагу, цей макрос відрізняється від стандарту тим, що назначає діалогу відкриття сторінки один (**oDialog2.Model.Step=1**). Тому ця сторінка буде виглядати так:



Завдання 4. Створіть макрос для переходу на наступну сторінку.

Така програма може мати наступний вигляд:

Sub cmdNext

if oDialog2.Model.Step<= 2 Then

oDialog2.Model.Step= oDialog2.Model.Step+ 1

end if

End Sub

Завдання 5. Створіть макрос для переходу на попередню сторінку.

Така програма може мати аналогічний вигляд:

```
Sub cmdPrev
```

```
if oDialog2.Model.Step>1 Then
```

```
oDialog2.Model.Step= oDialog2.Model.Step- 1
```

```
end if
```

```
End Sub
```

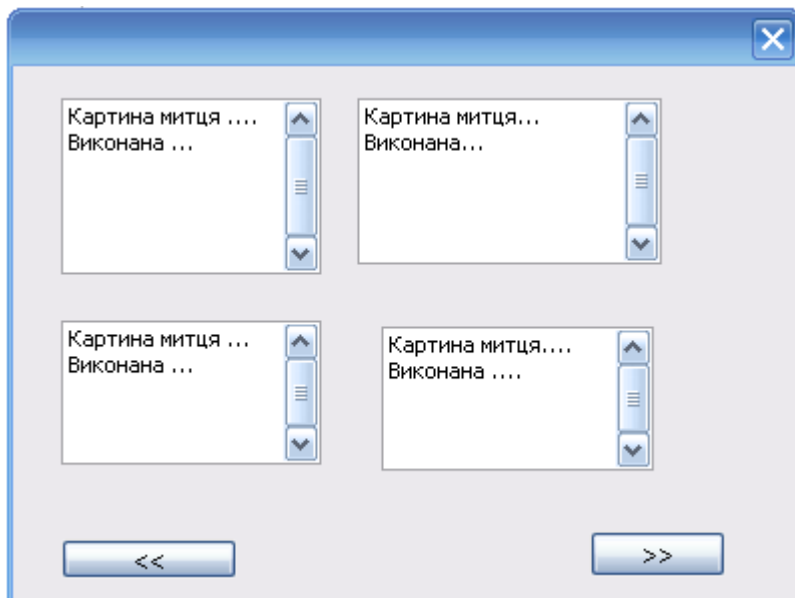
Зверніть увагу на те, що у макросах повинно бути виконано контроль переходу за межі можливого вибору сторінок.

Завдання 6. Зв'яжіть обидві макросі з відповідними кнопками та випробуйте роботу діалогу.

Якщо все зроблено добре, то друга сторінка може відобразитись, наприклад, так:



Третя сторінка може мати такий вигляд:



Створіть звіт з результатів виконання практичного заняття.

Практичне заняття № 5.
Тема: Обчислювальні діалоги.

Мета: засвоїти суть організації інтерфейсів з обчисленнями.

Завдання. Опанувати програмні засоби користувача щодо роботи з обчислювальними діалогами.

Зміст та послідовність виконання заняття:

Пояснити алгоритм для вирішення квадратного рівняння

$$ax^2 + bx + c = 0$$

з будь якими речовинними коефіцієнтами у речовинному полі чисел. Обчислювальні завдання виконуються за відповідними обчислювальними алгоритмами. Алгоритм для вирішення квадратного рівняння представлено нижче.

Згідно з визначенням ми маємо дійсне квадратне рівняння лише тоді, коли $a \neq 0$. якщо це не так, то рівняння вироджене, воно не вирішується, а користувачеві надається відповідне повідомлення.

Згідно з математичною формулою, маємо речовинні корені квадратного

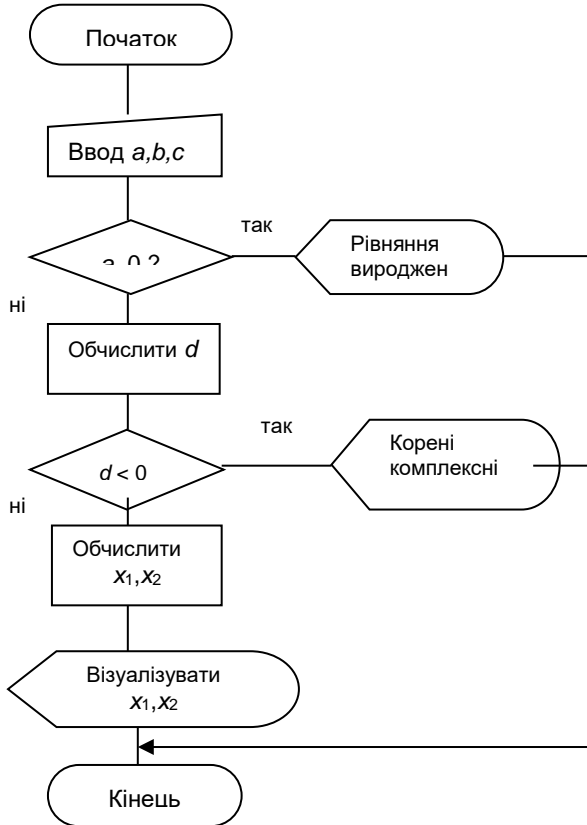
рівняння тоді і тільки тоді, коли дискримінант

$$d = b^2 - 4ac \geq 0$$

У цьому випадку для коренів рівняння маємо формули:

$$x_1 = \frac{-b + \sqrt{d}}{2a}; \quad x_2 = \frac{-b - \sqrt{d}}{2a}$$

У інших випадках корені будуть комплексними, що треба визначити повідомленнями.



Завдання 2.
Створіть у середовищі будь якого документа діалог. Розташуйте у полі цього діалогу надписи, текстові поля та кнопку так, щоб діалог мав такий вигляд, як на малюнку нижче.

Завдання 3. Створіть макрос, який буде розгортати діалог у середовищі документа, що було створено раніше. Для цього

поверніться на вкладку *Модули* вікна *Управление макросами LibreOffice Basic* і натисненням клавіші *Новый модуль* створить у своєму документі новий модуль, наприклад з ім'ям *Модуль1*. У тілі цього модуля запишіть інструкції розгортання діалогу, що наведено нижче. Вони стандартні.

Dim oDialogN As Object

'Замість N всюди 'підставте номер свого діалогу!

Sub DialogNShow

DialogLibraries.LoadLibrary("Standard")

oDialogN = CreateUnoDialog(DialogLibraries.Standard .DialogN)

oDialogN.Execute()

End Sub

Запустить процедуру **DialogNShow** на виконання та упевніться у тому, що його завантажено. Упевніться також у тому, що діалог поки ще не виконує функцій обчислення коренів. Дійсно, при роботі з діалогом видно, що всі дані у вікнах відображаються, але кнопка ОК мертва.

Завдання 4. Запрограмуйте кнопку ОК. Визначте ім'я відповідної процедури, скажемо QD (The quadratic decision) та оголосить усі дані, що потрібні на її виконання. (Дивись алгоритм розв'язання завдання, що розглянуто у завданні).

Таким чином маємо початок процедури з оголошеними даними:

Sub QD

Dim a As Single

Dim b As Single

Dim c As Single

Dim D As Single

Dim x1 As Single

Dim x2 As Single

Тепер треба отримати доступ до усіх текстових вікон діалогу. Це можна зробити, записав далі наступну послідовність інструкцій:

Dim oTextField1 As Object

Dim oTextField2 As Object

Dim oTextField3 As Object

Dim oTextField4 As Object

Dim oTextField5 As Object

```

oTextField1 = oDialog1.GetControl("TextField1")
oTextField2 = oDialog1.GetControl("TextField2")
oTextField3 = oDialog1.GetControl("TextField3")
oTextField4 = oDialog1.GetControl("TextField4")
oTextField5 = oDialog1.GetControl("TextField5")

```

Далі скачаємо дані з текстових вікон у змінні програми обчислення коренів. Для цього продовжимо текст програми:

```

a= oTextField1.text
b= oTextField2.text
c= oTextField3.text

```

Тепер можна проаналізувати значення коефіцієнта **a** та дискримінанту **D**. Згідно з алгоритмом вирішення рівняння маємо:

```

If a = 0 then
    MsgBox("Введіть коефіцієнт a не рівним нулю")
    Exit Sub 'Закінчуємо процедуру
Else
    D= b^2-4*a*c
    If D<0 then
        MsgBox("Корені комплексні")
        Exit Sub 'Закінчуємо процедуру
    Else

```

Тут уже маємо дані, що уведено коректно, тому далі пишемо:

```

    X1= (-b+sqr(D))/(2*a)
    X2= (-b-sqr(D))/(2*a)
    oTextField4.text=X1
    oTextField5.text=X2
end if

```

```

end if

```

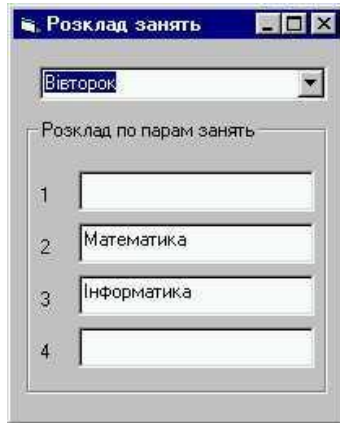
Закінчуємо програму.

```

End Sub

```

Завдання 5. Щоб ця програма спрацювала назначте процедуру **QD** кнопці ОК. Для цього у вікні проектування діалогу розкрийте вікно властивостей кнопки ОК, а потім перейдіть на вкладку *Подія*. Далі клацніть кнопку *Виконати дію* та й виконайте відповідне призначення за заданою послідовністю вікон.



Завдання 6. Виконайте верифікацію діалогу. Можна рахувати, що програма виконана вірно, якщо вона витримує наступні тести:

1. Для значень $a = 0$, b , c — які завгодно, при запуску програми повинно бути повідомлення, що рівняння вироджене. При цьому текстові вікна коренів повинні бути очищеними.

2. Для значень $a = 1$, $b = 1$, $c = 1$ у текстові вікна коренів повинно бути видано повідомлення: “Корені комплексні”

3. Для значень $a = 1$, $b = 4$, $c = 4$ значення коренів мають бути такі: $x_1 = -2$; $x_2 = -2$.

4. Для значень $a = 1$, $b = -9$, $c = 20$ маємо відповідні корені: $x_1 = 5$; $x_2 = 4$.

Якщо програма не витримала хоча б одного із тестів, її треба проаналізувати на логічні помилки, а потім — ці помилки усунути.

Створіть звіт з результатів виконання практичного заняття

Практичне заняття № 6.

Тема: Довідкові діалоги.

Мета: навчитися типовим прийомам роботи з довідковими діалогами.

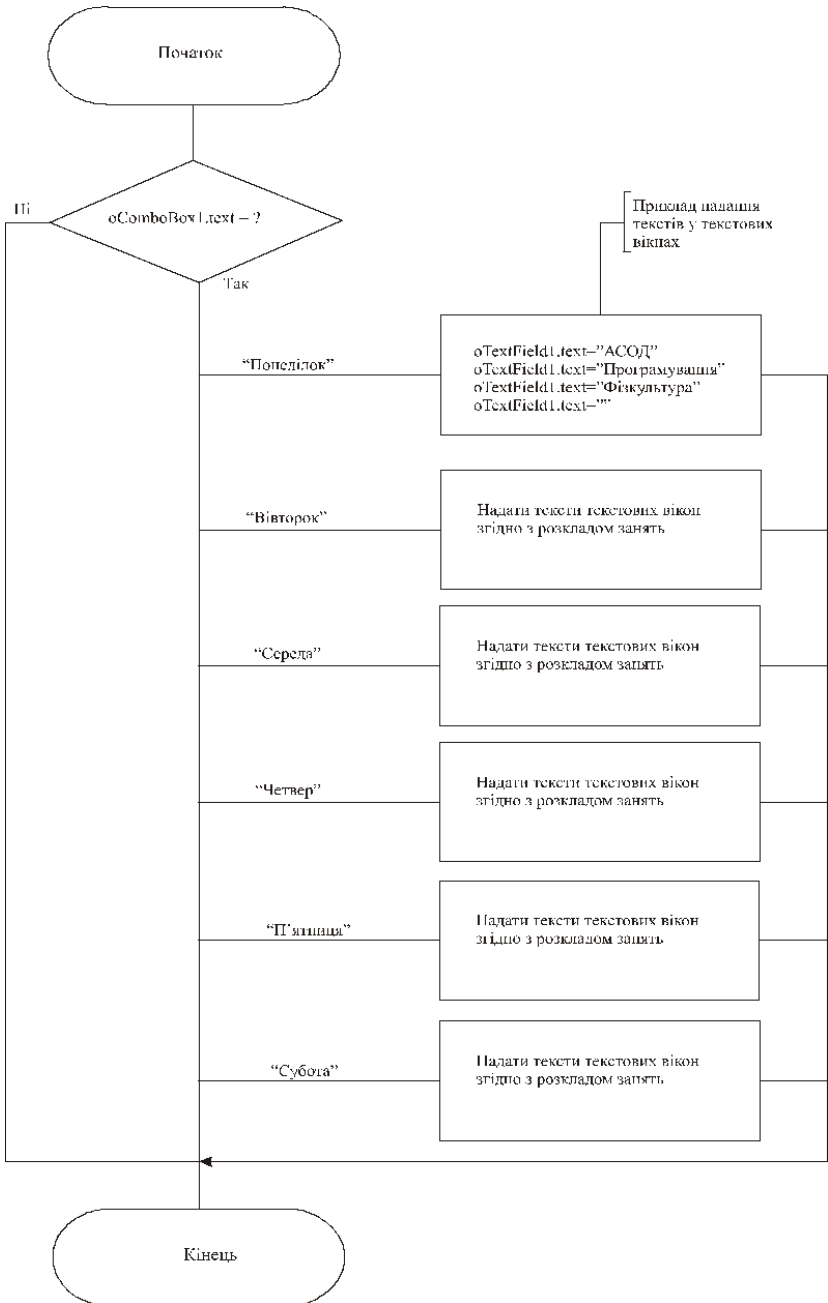
Завдання. Опанувати програмні засоби користувача щодо створення довідкових діалогів.

Зміст та послідовність виконання заняття:

Завдання 1. Уявити алгоритм створення довідкового діалогу на прикладі «Розклад занять на тиждень», зовнішній вигляд якого дано на малюнку нижче.

Для вирішення подібних завдань часто застосовуються розгалуження. Для наповнення текстових полів довідковою

інформацією можна користуватися інструкцією умов If Then ...Else, однак у більшості випадків буде більш доцільною інструкція Select Case [*Довідка/Макроси та програмування/ Довідник команд/ Функції, інструкції та оператори часу виконання/ Управління виконанням програми/ Інструкції умов/ Select ... Case*]. Нижче показано, як ця інструкція працює. Розгляньте її уважно.



Завдання 2. Створіть діалог, що реалізує довідковий алгоритм, якого надано вище. Для створення діалогу застосуйте поле зі списком, чотири текстових вікна та чотири надписи.

Завдання 3. Створіть макрос, який буде розгортати діалог у середовищі документа, що було створено раніше. Для цього поверніться на вкладку *Модулі* вікна *Управління макросами LibreOffice Basic* і натисненням клавіші *Новий модуль* створіть у своєму документі новий модуль, наприклад з ім'ям *Модуль1*. У тілі цього модуля запишіть інструкції розгортання діалогу, що наведено нижче. Вони можуть бути такими:

Dim oDialogN As Object

'Замість N всюди 'підставте номер свого діалогу!

Sub DialogNShow

DialogLibraries.LoadLibrary("Standard")

oDialogN = CreateUnoDialog(DialogLibraries.Standard.DialogN)

oDialogN.Execute()

End Sub

Запустить процедуру **DialogNShow** на виконання та упевніться у тому, що діалог завантажено. Упевніться також, що діалог поки ще не виконує функцій надання довідки.

Завдання 4. Завантажте у процедурі розгортання діалогу елемент управління *Поле зі списком (oComboBox1)* усіма необхідними даними. Попередньо рекомендується оголосити усі об'єктні змінні. Приклад дано нижче.

Dim oDialog1 As Object

Dim oComboBox1 As Object

Dim oTextField1 As Object

Dim oTextField2 As Object

Dim oTextField3 As Object

Dim oTextField4 As Object

Sub Dialog1Show

DialogLibraries.LoadLibrary("Standard")

oDialog1=CreateUnoDialog(DialogLibraries.Standard.Dialog1)

oComboBox1=oDialog1.GetControl("ComboBox1")

oComboBox1.AddItem("Понеділок",1)

```
oComboBox1.AddItem("Вівторок",2)
```

```
'Далі продовжити аналогічно для всіх днів тижня
```

```
'Запуск діалогу на виконання:
```

```
oDialog1.Execute()
```

```
End Sub
```

Завдання 4. Створіть процедуру надання довідки. Для цього можна орієнтуватись на наступний примірник:

```
Sub schedule
```

```
oComboBox1 = oDialog1.GetControl("ComboBox1")
```

```
oTextField1 = oDialog1.GetControl("TextField1")
```

```
oTextField2 = oDialog1.GetControl("TextField2")
```

```
oTextField3 = oDialog1.GetControl("TextField3")
```

```
oTextField4 = oDialog1.GetControl("TextField4")
```

```
Select Case oComboBox1.text
```

```
Case is="Понеділок"
```

```
oTextField1.text="АСОД"
```

```
oTextField2.text="Математика"
```

```
oTextField3.text="Фізкультура"
```

```
oTextField4.text=""
```

```
Case is="Вівторок"
```

```
oTextField1.text=""
```

```
oTextField2.text="Математика"
```

```
oTextField3.text="Інформатика"
```

```
oTextField4.text=""
```

```
'Далі продовжить на всі дні тижня аналогічно
```

```
Case Else 'На усе, що не визначене у oComboBox1.text
```

```
oTextField1.text=""
```

```
oTextField2.text=""
```

```
oTextField3.text=""
```

```
oTextField4.text=""
```

```
End Select
```

```
End Sub
```

Випробуйте створену процедуру клацанням мишки у полі з списком. Якщо усе зроблено вірно, то у текстових вікнах буде

відображено потрібну інформацію. У протилежному випадку налагодьте діалог.

Завдання 5. Повністю самостійно створити подібний до попереднього діалог, який при виборі виду навчальних занять у комбінованому списку “Види занять” відображає відповідні номери тижнів на поточний семестр у текстовому вікні.

Для деякого потоку студентів затверджено такий графік навчального семестру:

«Аудиторні заняття» — проводяться з першого по одинадцятий тиждень,

«Виробнича практика» — з дванадцятого по шістнадцятий тиждень,

«Заліковий тиждень» — сімнадцятий,

«Іспити» — з вісімнадцятого по двадцять четвертий тижні.

Завдання 6. Випробуйте діалог, створений у попередньому завданні, та представте результати викладачеві.

Створіть звіт з результатів виконання практичного заняття.

Практичне заняття № 7.

Тема: Діалоги сортувань.

Мета: засвоїти технологію організації та програмування інтерфейсів сортувань даних.

Завдання. Засвоїти застосування елементів управління в діалогах при організації та програмуванні інтерфейсів сортувань даних.

Зміст та послідовність виконання заняття:

Завдання 1. Поясніть постановку завдання сортувань даних. Таке завдання часто має місце тоді, коли структуровані дані треба представити підпорядкованими по якомусь визначеному критерію (ключу). Хай інтерфейс діалогу має такий вигляд, як дано на примірнику нижче.

Прізвище та ініціали	Посада	Тел. мобільний
Петров І. Г.	Нач. віддділу	097-572-56-78
Петрова Т. П.	Зам. нач. віддділу	067-432-75-12
Жуков Г. Г.	Ст. менеджер	050-223-67-43

Діалог створено з трьох кнопок, що розташовані над списками. При натисненні на кнопку повинно виконуватись сортування по ключу. Сенс ключа визначено надписом кнопки. При цьому структура кожного із кортежів повинна зберегтись. Дані для кортежів, як правило, дістаються із текстових файлів, а потім при роботі інтерфейсу пересуваються у списки. Ми будемо емалювати процес наповнення списків даними завантаженням їх на етапі ініціалізації діалогу.

Завдання 2. У середовищі будь якого з документів створити пустий діалог. Для цього поверніться на вкладку *Діалоги* вікна *Управління макросами* і натисненням клавіші *Новий діалог* створіть його у своєму документі. Розмістіть кнопки на діалогі і порівняйте їх по висоті засобами властивостей. Розмістіть списки на діалогі і також вирівняйте їх засобами властивостей. При цьому треба дотримуватися відповідності по ширині кнопок і висоті списків.

Завдання 3. Створіть макрос, який буде розгортати діалог у середовищі документа, який було створено раніше. Для цього поверніться на вкладку *Модулі* вікна *Управління макросами* і натисненням клавіші *Новий модуль* створіть у своєму документі новий модуль, наприклад з ім'ям *Модуль1*. У тілі цього модуля запишіть процедуру розгортання діалогу, що наведено нижче. Текст цієї процедури буде відрізнятись від стандартного тим, що маєтсья виклик **AddDate()** завантаження списків даними.

Dim oDialogN AS Object

' **Замість символу N** задайте номер діалогу, наприклад: **1**

```
Sub DialogNShow
```

```
DialogLibraries.LoadLibrary( "Standard" )
```

```
oDialogN = CreateUnoDialog( DialogLibraries.Standard .DialogN )
```

```
oDialogN.Execute()
```

```
AddDate()
```

```
End Sub
```

Завдання 4. Створіть процедуру **AddDate()** завантаження списків даними. Примірник такої процедури дано нижче. Головне, що треба зробити цієї процедурі, це дістатися моделей списків, а потім завантажити дані з клавіатури у списки.

```
Sub AddDate()
```

```
' Дістаємося моделей списків.
```

```
oListBox1=oDialog1.GetControl("ListBox1")
```

```
oListBox2=oDialog1.GetControl("ListBox2")
```

```
oListBox3=oDialog1.GetControl("ListBox3")
```

```
' завантажуюмо дані.
```

```
oListBox1.AddItem("Петров І. Г.",0)
```

```
oListBox2.AddItem("Нач. відділу",0)
```

```
oListBox3.AddItem("097-572-56-78",0)
```

```
oListBox1.AddItem("Петрова Т. П.",1)
```

```
oListBox2.AddItem("Зам. нач. відділу",1)
```

```
oListBox3.AddItem("067-432-75-12",1)
```

```
oListBox1.AddItem("Жуков Г. Г.",2)
```

```
oListBox2.AddItem("Ст. менеджер",2)
```

```
oListBox3.AddItem("050-223-67-43",2)
```

```
' Таким чином наберіть десь з десяток записів о службовцях.
```

```
End sub
```

Випробуйте процедуру розгортання діалогу. Наповнення списків даними повинно бути виконано автоматично після запуску розгортання діалогу.

Завдання 4. Створіть для кнопки *Прізвище та ініціали* макрос сортувань по ключу. Нижче дано примірник такого макросу. Кожний з підрозділів прокоментовано.

```
Sub Sorting()
```

```
Dim i As Integer ' Оголошуємо робочі змінні
```

```

Dim j As Integer
Dim N as Integer
Dim ps As String
Dim Fio() as String ' Оголошуємо робочі масиви.
Dim Stat() as String
Dim Tel() as String
' У змінній N маємо кількість елементів у списку:
N=oListBox1.getItemCount()
' Уточнюємо кількість елементів у масивах:
redim FIO(N) as String
redim stat(N) as String
redim Tel(N) as String
' Перепишуємо списки у масиви
For i= 1 to N
Fio(i) =oListBox1.getItem(i-1)
Stat(i)=oListBox2.getItem(i-1)
Tel(i)=oListBox3.getItem(i-1)
Next i
' Виконуємо сортування масивів методом бульбашки (гл. конспект).
For j = N-1 To 1 Step -1
For i = 1 To j
If Fio(i) > Fio(i+1) Then
    ps = Fio(i)
    Fio(i) = Fio(i+1)
    Fio(i+1)=ps
    ps = Stat(i)
    Stat(i) = Stat(i+1)
    Stat(i+1)=ps
    ps = Tel(i)
    Tel(i) = Tel(i+1)
    Tel(i+1)=ps
End If
Next i
Next j
' Очищуємо списки, починаючи з позиції 0, з усіма N записами.

```



```
oListBox1.removeItem(0,N)
```

```
oListBox2.removeItem(0,N)
```

```
oListBox3.removeItem(0,N)
```

' Заповнюємо списки масивами, що відсортовані:

```
For i= 1 to N
```

```
oListBox1.addItem(Fio(i),i-1)
```

```
oListBox2.addItem(Stat(i),i-1)
```

```
oListBox3.addItem(Tel(i),i-1)
```

```
Next i
```

```
End Sub
```

Розмістіть цей макрос у тому ж самому модулі, де розміщено попередні два макроси. Зв'яжіть його з подією *Виконати дію* даної кнопки. Випробуйте роботу діалогу на рівні кнопки, що запрограмована.

Завдання 5, 6 Створіть для решти двох кнопок аналогічні макроси та випробуйте їх. При випробуваннях кнопок сортування даних повинно відбуватись по ключу, що визначено підписом відповідної кнопки.

Створіть звіт з результатів виконання практичного заняття.

Практичне заняття № 8.

Тема: Діалоги підключувань файлів.

Мета: засвоїти технологію організації та програмування інтерфейсів підключувань файлів даними.

Завдання. Опанувати методику організації та програмування інтерфейсів підключувань файлів даними.

Зміст та послідовність виконання заняття.

Завдання 1. Поясніть постановку завдання. Подібні завдання можуть мати місце тоді, коли немає необхідності заводити базу даних, але треба виконувати збереження структурованої інформації у файлах можливо великого обсягу.

Хай, наприклад, маємо завдання, щоб кожного дня записувати витрати по статтям у файл, записи котрого мають таку структуру:

Дата| Найменування статті витрат| Сума витрат|

Треба створити простий інтерфейс наповнення файлу даними.

Завдання 2. Створіть діалог наповнення файлу даними, примірник якого дано нижче. Цей діалог має такі поля вводу даних: поле дати, поле со списком та поле валюти.

У полі валюти слід задати властивість *Символ валюти* як Грн. Поля написів застосуйте для найменування усіх визначених полів. Призначення кнопки *Добавити запис* розглядайте у тому сенсі, щоб писати кожен запис у кінець файлу, якого обрано для цієї справи.

The image shows a standard Windows-style dialog box with a close button (X) in the top right corner. It contains three input fields and one button:

- The first field is labeled "Дата" (Date) and is an empty text box.
- The second field is labeled "Найменування статті витрат" (Expense description) and is a dropdown menu with a small downward arrow on the right side.
- The third field is labeled "Сума" (Sum) and is an empty text box.
- At the bottom right, there is a button labeled "Добавити запис" (Add record).

Завдання 3. Створіть макрос, ініціалізації діалогу. В цей макрос крім стандартних інструкцій доцільно включити інструкції заповнення статей витрат та відображення поточної дати. Тоді користувачеві буде досить: вибрати статтю витрат з поля зі списком, записати у поле валюти суму витрат та клацнути кнопку *Добавити запис*. Примірник такого макросу дано нижче:

Dim oDialog1 AS Object

Sub Dialog1Show

DialogLibraries.LoadLibrary("Standard")

oDialog1 = CreateUnoDialog(DialogLibraries.Standard.Dialog1)

```
'MsgBox(oDialog1.model.DBG_Properties)
oDialog1.model.title="Запис даних у файл"
oDateField1=oDialog1.GetControl("DateField1")
oDateField1.Model.Date= CDateToIso(Date())
oComboBox1=oDialog1.GetControl("ComboBox1")
oComboBox1.AddItem("Продукти харчування",0)
oComboBox1.AddItem("Одяг",1)
oComboBox1.AddItem("Обувка",2)
oComboBox1.AddItem("Предмети туалету",3)
oDialog1.Execute()
End Sub
```

Розмістіть даний макрос в відповідному модулі та випробуйте його. При випробуванні макросу повинна бути відображена поточна дата у відповідному полі, а при клацанні кнопки поля зо списком — відбуватися поява меню вибору статей витрат. Кнопка пересувань кортежу у файл поки ще не повинна діяти.

Завдання 4. Створіть макрос, що підтримує роботу кнопки *Добавити запис*. Ця кнопка повинна забезпечити відкриття текстового файлу, що завдано своїм маршрутом та записати у цей файл кортеж: поточну дату, назву статті витрат і суму витрат.

Рекомендується виконати це завдання так. Спочатку створіть у своєму каталозі пустий текстовий файл засобами *Блокноту*. Потім скопіюйте маршрут до цього файлу з вікна властивостей даного *Блокнота* у буфер обміну, а потім вставте рядок буфера обміну у змінну, що буде визначати даний маршрут. Надайте до маршруту ще й ім'я файлу, що було визначено при його створенні.

Хай змінна, що зберігає повний маршрут буде мати ім'я **aFile**. Тоді у деякому комп'ютері може бути таке визначення цієї змінної:

```
aFile = "D:\D_Мои документи\Витрати.txt"
```

З подальшим створенням макросу можна ознайомитися з наступних коментарів:

```
Sub InFile
```

```
aFile = "D:\D_Мои документи\Витрати.txt"
```

```
iNumber = Freefile 'Створюємо дискриптор файлу
```

Open aFile For Append As #iNumber'Відкриваємо файл для поповнення

'Дістаємося до моделей елементів управління

oDateField1=oDialog1.GetControl("DateField1")

oComboBox1=oDialog1.GetControl("ComboBox1")

oCurrencyField1=oDialog1.GetControl("CurrencyField1")

'Читаємо дані з елементів управління

Dim dd As String

Dim State as String

dim Cost As String

dd=oDateField1.model.Date()

dd=dd+"|"

State=oComboBox1.model.text

State=state+"|"

Cost=oCurrencyField1.model.value

' Пишемо дані у файл

print #iNumber;dd;State;Cost

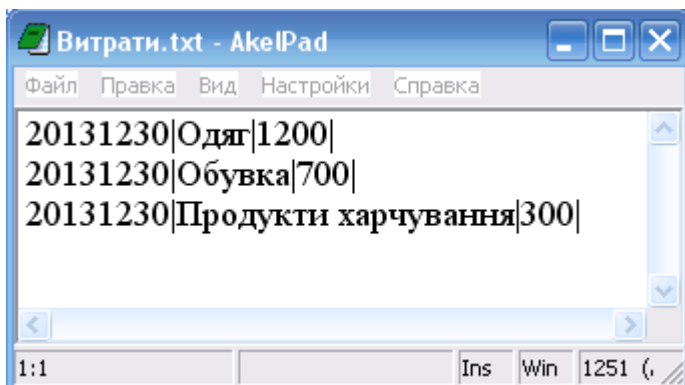
' Закриваємо файл

close #iNumber

end sub

Завдання 5. Зв'яжіть макрос **InFile** з подією *Виконати дію* кнопки *Добавити запис*.

Завдання 6. Випробуйте діалог у цілому. Для цього кілька разів поспіль введіть необхідні дані та кожного разу клацайте кнопку *Добавити запис*. Відкрийте текстовий файл системними засобами. Повинно відобразитись щось подібне до цього:



Якщо випробування на деякому кроку не вдалося, знайдіть помилки у макросах або у властивостях елементів управління та виправте їх.

Зверніть ще увагу на те, як розділено елементи кортежів. Такий прийом надає можливості автоматизувати пошук у файлі. Створений файл можна редагувати у ручну, якщо дозволено у процедурі **InFile**, але можна і заборонити таку можливість. Обов'язково збережіть текстовий файл, бо він знадобиться на подальшому заняттю.

Створіть звіт з результатів виконання практичного заняття.

Практичне заняття № 9.

Тема: Діалоги пошуку у текстових файлах.

Мета: засвоїти організацію та програмування пошукових інтерфейсів.

Завдання. Опанувати методику організацію та програмування пошукових інтерфейсів.

Зміст та послідовність виконання заняття.

Завдання 1. Поясніть постановку завдання. Подібні завдання можуть мати місце тоді, коли недоцільно заводити базу даних, чи електронну таблицю, але треба виконувати пошук структурованої інформації у текстових файлах, можливо великого обсягу.

Хай, наприклад, маємо файл, де кожного дня записувались витрати по статтям. У цьому файлі кожна із записів має таку структуру:

Дата| Найменування статті витрат| Сума витрат|

Треба створити інтерфейс пошуку, що обчислює, наприклад, загальні витрати за певний період часу. Для цього скористуйтеся файлом, якого було створено на попередньому занятті.

Завдання 2. Створіть діалог пошуку. Хай цей діалог має такий вигляд, як подано на малюнку нижче. У даному випадку діалог доцільно створити у той же самій бібліотеці, що і на попередньому занятті. Тому ім'я діалогу обрано, як **Dialog2**.

Для вводу дат передбачте вікна дат. У цих вікнах встановіть властивості *Розгорнутий* у становище *да*. Тоді повинна з'явитися кнопка розгортання, що надасть можливість користуватись календарем для вибору дат. Поле валюти передбачте для відображення цієї суми, що шукається. Вибір кнопки особливостей не має.

Завдання 3. Створіть макрос, ініціалізації діалогу. Цей макрос достатньо наповнити стандартними інструкціями.

Dim oDialog2 AS Object

Sub Dialog2Show

```

DialogLibraries.LoadLibrary( "Standard" )
oDialog2 = CreateUnoDialog( DialogLibraries.Standard.Dialog2 )
MsgBox(oDialog1.model.DBG_Properties)
oDialog2.model.title="Пошук даних у файлі"
oDialog2.Execute()
End Sub

```

Випробуйте макрос з тим, щоб упевнитися у нормальній роботі полів дат. Дати повинні відображатися спочатку на календарях, а після вибору дати — у вікнах..

Права частина діалогу реагувати на клацання мишкою не повинна.

Завдання 4. Створіть макрос, що підтримує кнопку пошуку. Цей макрос повинен забезпечити читання кожного рядка з текстового файлу, розбивати рядок на три частини: дату, статтю витрат та обсяг витрат. Перевіряти межі дат, що прочитано і , якщо ці межі відповідають заданим, то накопичувати суму витрат. Коли усі рядки буде прочитано, загальна сума повинна відобразитись у відповідному полі.

Примірник тексту такого макросу надано нижче.

```

Sub FindFile
Dim dd1 as Long
Dim dd2 as long
'дістанемоь моделей елементів управління, що потрібні
oDateField1=oDialog2.GetControl("DateField1")
oDateField2=oDialog2.GetControl("DateField2")
oCurrencyField1=oDialog2.GetControl("CurrencyField1")
'межі термінів часу збережемо у змінних
dd1=oDateField1.model.Date()
dd2=oDateField2.model.Date()
'якщо маємо помилку, то вихід із макросу
if dd2<dd1 then
Msgbox("Не вірно вказано дати")
exit sub
end if
'готуємо читання із файлу

```

```

aFile = "D:\D_Мои документы\Витрати.txt"
iNumber = Freefile ' Створюємо дескриптор файлу
Open aFile For Input As #iNumber'Відкриваємо файл для читання
'оголошуємо допоміжні змінні
Dim FileString as String
dim S as Currency
dim cs as Currency
reDim StrArr(3) as String
'створюємо пошук: поки не кінець файлу, шукати
Do While not Eof(iNumber)
Line Input #iNumber,FileString 'весь рядок читаємо у змінну
FileString
StrArr=split(FileString,"|",3) 'розбиваємо рядок на три підрядки
'якщо знаходимось у завданих межах, то накопичуємо суму
if (CLng(StrArr(0))>= dd1) And (CLng(StrArr(0)))<=dd2 then
cs = Cdbl(StrArr(2))
S=S+ cs
end if
Loop 'кінець циклу пошуку
oCurrencyField1.model.value=S' виводимо суму у валютне вікно
close #iNumber' закриваємо файл
end Sub

```

Налагодьте такий макрос та випробуйте його.

Створіть звіт з результатів виконання практичного заняття.

Самостійні роботи

Самостійна робота № 1.

Тема: Поняття інтерфейсу користувача і основні вимоги до нього.

Мета: з'ясувати суть інтерфейсу користувача і основні вимоги до нього.

Завдання:

Створити звіт з самостійної роботи, де відобразити відповіді на питання:

1. Що таке інтерфейс користувача в умовах сучасного розвитку комп'ютерної техніки?
2. У чому полягають основні принципи розробки інтерфейсів користувача?
3. У чому є суть стандартизації інтерфейсів користувача?
4. Які можна визначити основні етапи проектування інтерфейсів користувача?
5. Що таке вікно діалогу і які елементи у ньому зосереджуються?

Самостійна робота № 2.

Тема: Основні зведення про проектування інтерфейсів користувача. Особливості проектування графічного інтерфейсу.

Мета: З'ясувати суть зведення про проектування інтерфейсів користувача.

Завдання на відпрацювання конспекту лекцій і літератури:

Створити звіт з самостійної роботи, де відобразити конспективні відповіді на питання:

1. Поясніть що таке форма і як вона пов'язується з моделлю елемента управління.
2. Які аспекти має модель елемента управління?
3. Як дістатися моделі елемента управління форми програмним шляхом?
4. Як прочитати властивості елемента управління на формі програмним шляхом?
5. Як можна змінити властивості моделі елемента управління у робочому становищі форми?

6. Як дістатися графічних параметрів елемента управління та встановити їх програмними засобами?

Самостійна робота № 3.

Тема: Елементи управління, як діалогові об'єкти. Вікна, піктограми та їх проектування.

Мета: З'ясувати структуру елементів управління, як діалогових об'єктів, а також вікон, піктограм та їх проектування.

Завдання на відпрацювання конспекту лекцій і літератури:

Створити звіт, де відобразити конспективні відповіді на наступні питання:

1. Як створювати прості діалоги?
2. Як працювати з властивостями?
3. Як коригувати програмно положення та розмір елементів управління?
4. Як працювати програмно з фокусом об'єкта?
5. Як додавати нові елементи управління?

Самостійна робота № 4.

Тема: Обчислювальні діалоги. Типи та різновиди меню. Кнопки, перемикачі, прапорці.

Мета: поглибити поняття обчислювальних діалогів, робота з типами та різновидами меню, додавання кнопок, перемикачів, прапорців.

Завдання на відпрацювання конспекту лекцій і літератури:

Створити звіт з самостійної роботи, де відобразити конспективні відповіді на наступні питання:

1. Що таке багатосторінковий діалог?

2. Як виконується проектування багатосторінкового діалогу у Office.org?
3. Якими засобами підтримується управління сторінками діалогу?
4. Як контролюється перебування у завданих межах сторінок діалогу?
5. Які засоби користування має графічне поле, як елемент управління?

Самостійна робота № 5.

Тема: Списки. Текстові поля.

Мета: засвоїти поняття щодо списків та текстових полей на діалозі.

Завдання на відпрацювання конспекту лекцій і літератури:

Створити звіт з самостійної роботи, де відобразити конспективні відповіді на наступні питання:

1. Яке призначення мають текстові поля?
2. Як відбувається обмін даними між текстовими полями та програмами їх обслуговування?
3. Як відображаються у текстових полях дані, що не можуть бути визначені?
4. Яке призначення мають кнопки?
5. Як програмується кнопка на виконання певної дії?

Самостійна робота № 6.

Тема: Засоби угруповання прокрутки та індикації. **Проектування засобів підтримки користувача. Методика проектування інтерфейсів: від загального до окремого**

Мета: поглибити знання відносно засобів групування, прокрутки та індикації. а також проектування засобів підтримки користувача і методики проектування інтерфейсів: від загального до окремого.

Завдання на відпрацювання конспекту лекцій і літератури:

Створити звіт з самостійної роботи, де відобразити конспективні відповіді на наступні питання:

1. Що таке довідковий діалог і яку типову структуру він має?
2. Як завантажити дані у поле зі списком?
3. Як прочитати дані з поля зі списком?
4. Як дізнатися індексу елемента у списку?
5. Яку семантику має конструкція Select Case?
6. Яку загальну структуру має інтерфейс сортувань?
7. Як організовано алгоритм сортувань бульбашками?
8. Як і яка властивість застосовується для запису даних у список?
9. Яку роль відіграє масив даних в умовах сортувань?
10. Що таке динамічний масив і як він підтримується програмно?

11. Яка типова структура інтерфейсів підключення файлів даних?
12. Які переваги може мати приватний інтерфейс підключення файлів і коли його доцільно застосовувати?
13. Які програмні засоби застосовуються для роботи з файлами?
14. Як поділяти дані у кортежу рядка, щоб їх можна було потім упізнати структурно?

15. Чи може і коли бути доцільним створення пошукових засобів у приватних інтерфейсах користувача?
16. Які програмні засоби застосовуються для читання інформації із особистих файлів користувача?
17. Яка функція застосовується для розділення даного рядка на складові?
18. Як організувати читання з файлу від початку і до кінця?

Методи навчання

Пояснювально-ілюстративний, частинно-пошуковий (евристичний), репродуктивний, набуття знань, практичний, словесний, наочний, формування умінь і навичок, використання знань творча діяльність, закріплення, перевірка знань, умінь і навичок.

Форми контролю

№ з/п	Вид контролю	Методи контролю
1.	Поточний	усне опитування, перевірка якості виконання практичних завдань
2.	Періодичний	контрольні тестові роботи після завершення розділів,
3.	Самостійна робота	перевірка якості виконання завдань самостійної роботи
4.	Підсумковий контроль	іспит

Поточний контроль здійснюється у формі усного опитування, контрольних тестових робіт після завершення розділів, а також якості виконання завдань самостійної роботи та практичних завдань, тестування.

Підсумковий контроль здійснюється у формі іспиту

Відпрацювання пропущених занять здійснюється за узгодженням між студентом та викладачем.

Перезарахування результатів вивчення навчальної дисципліни відбувається у присутності комісії за умови доведення студентом/студенткою того рівня оцінки, на якій він/вона претендує.

Поточне тестування та самостійна робота						Сума
Розділ 1		Розділ 2				
T1	T2	T3	T4	T5	T6	100
16	16	16	16	16	20	

Розподіл балів, які отримують здобувачі освіти

T1, T2, ..., T6 — теми розділів.

Розподіл балів за формами контролю.

- відвідування лекцій (пара) — 2; ...6 пар — до 12 балів
- виконання практичного завдання – 1-6; 11 практичних – 55 бал
- усне опитування — 1; ... 3 усних опитування — до 3 балів
- письмове опитування — до 10 балів
- виконання самостійного завдання — 1-2; 13 самостійних завдань — до 26 балів

Можлива кількість балів, набраних до підсумкової форми контролю:
80 балів

залік — 20 балів

Шкала оцінювання: національна та ECTS

Сума балів за всі види навчальної діяльності	Оцінка ECTS	Оцінка за національною шкалою	
		для екзамену, курсового проекту (роботи), практики	Для заліку
90-100	A	відмінно	зараховано
82-89	B	добре	
74-81	C		
64-73	D	задовільно	
60-63	E		

35-59	FX	незадовільно з можливістю повторного складання	не зараховано з можливістю повторного складання
0-34	F	незадовільно з обов'язковим повторним вивченням дисципліни	не зараховано з обов'язковим повторним вивченням дисципліни

Рекомендована література

Обов'язкова

1. OpenOffice.org 3.2 BASIC Guide. URL: https://wiki.openoffice.org/w/images/c/c1/BasicGuide_OOo3.2.0.pdf (дата звернення: 25.01.2024).

2. Andrew D. Pitonyak OpenOffice.org Macros Explained OOME Third Edition. URL: https://www.pitonyak.org/OOME_3_0.pdf (дата звернення: 25.01.2024).

3. OpenOffice.org 3.1 Developer's Guide. URL: https://wiki.openoffice.org/w/images/d/d9/DevelopersGuide_OOo3.1.0.pdf (дата звернення: 25.01.2024).

4. Learn OpenOffice.org Spreadsheet Macro Programming OOoBasic and Calc Automation Guide. URL: <https://www.pdfdrive.com/learn-openofficeorg-spreadsheet-macro-programming-ooobasic-and-calc-automation-a-fast-and-friendly-tutorial-to-writing-macros-and-spreadsheet-applications-e184846880.html> (дата звернення: 25.01.2024).

Допоміжна

5. The Apache OpenOffice API Project / Керівництво розробника: сайт. URL: <http://www.openoffice.org/api/> (дата звернення: 25.01.2024).

6. The OpenOffice.org Documentation Project: сайт. URL: <http://www.openoffice.org/api/> . (дата звернення: 25.01.2024).

7. Index of /api/docs/common/ref/com/sun/star URL: <http://www.openoffice.org/api/docs/common/ref/com/sun/star/> (дата звернення: 25.01.2024).

Інформаційні ресурси

1. Бібліотека Харківської державної академії культури.
2. Бібліотека Харківського національного університету В. Н. Каразіна.
3. Харківська державна наукова бібліотека імені В. Г. Короленка.
4. Українська бібліотечна енциклопедія [Електронний ресурс].
– Режим доступу : <http://ube.nplu.org>. – Назва з екрану.

КОНТРОЛЬНІ ПИТАННЯ З ДИСЦИПЛІНИ

1. Що таке інтерфейс користувача в умовах сучасного розвитку комп'ютерної техніки?
2. У чому полягають основні принципи розробки інтерфейсів користувача?
3. У чому є суть стандартизації інтерфейсів користувача?
4. Які можна визначити основні етапи проектування інтерфейсів користувача?
5. Що таке вікно діалогу і які елементи у ньому зосереджуються?
6. Поясніть що таке форма і як вона пов'язується з моделлю елемента управління.
7. Які аспекти має модель елемента управління?
8. Як дістатися моделі елемента управління форми програмним шляхом?
9. Як прочитати властивості елемента управління на формі програмним шляхом?
10. Як можна змінити властивості моделі елемента управління у робочому становищі форми?
11. Як дістатися графічних параметрів елемента управління та встановити їх програмними засобами?
12. Що таке багатосторінковий діалог?
13. Як виконується проектування багатосторінкового діалогу у Office.org?
14. Якими засобами підтримується управління сторінками діалогу?

15. Як контролюється перебування у завданих межах сторінок діалогу?
16. Які засоби користування має графічне поле, як елемент управління на багатосторінковому діалозі?
17. Які основні типи діалогів застосовуються у інтерфейсах користувача?
18. Які риси має діалог обчислення?
19. Яке призначення мають текстові вікна у діалозі обчислень?
20. Як відбувається обмін даними між текстовими вікнами та програмами їх обслуговування?
21. Як відображаються у текстових вікнах дані, що не можуть бути визначені?
22. Яке призначення мають кнопки?
23. Як програмується кнопка на виконання певної дії?
24. Що таке довідковий діалог і яку типову структуру він має?
25. Як завантажити дані у поле зі списком?
26. Як прочитати дані з поля зі списком?
27. Як дізнатися індексу елемента у списку?
28. Яку семантику має конструкція Select Case?
29. Яку загальну структуру має інтерфейс сортувань?
30. Як організовано алгоритм сортувань бульбашками?
31. Яку роль відіграє масив даних в умовах сортувань?
32. Що таке динамічний масив і як він підтримується програмно?
33. Чи може і коли бути доцільним створення пошукових засобів у приватних інтерфейсах користувача?
34. Які програмні засоби застосовуються для читання інформації із особистих файлів користувача?
35. Як організувати читання з файлу від початку і до кінця?

ТЕХНОЛОГІЯ ПРОГРАМУВАННЯ ІНТЕРФЕЙСІВ

Програма та навчально-методичні матеріали до курсу
для здобувачів першого (бакалаврського) рівня вищої освіти зі
спеціальності 029 «Інформаційна, бібліотечна та архівна справа»
ОП «Інформаційна та документаційна діяльність»

Укладач:

Побіженко Ірина Олександрівна,
кандидат технічних наук, доцент, доцент кафедри цифрових
комунікацій та інформаційних технологій ХДАК

Друкується в авторській редакції

План 2024

Підписано до друку 24.04.2024 р. Формат 60x84/16
Гарнітура «Times». Папір для мн. ап. Друк ризограф.
Ум. Друк арк. 1,99. Обл.-вид. арк. 2,08. Тираж 100. Зам . №
ХДАК, 61057, м. Харків, Бурсацький узвіз, 4
Надруковано в лаб. множ. техніки ХДАК