

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЖИТОМИРСЬКИЙ ДЕРЖАВНИЙ УНІВЕРСИТЕТ ІМЕНІ ІВАНА ФРАНКА

Т.А. ВАКАЛЮК

Візуальне програмування

*навчально-методичний посібник
для студентів фізико-математичного факультету*

Житомир 2013

УДК 004.42+004.432.2

ББК 32.973.26-018.1

В14

*Затверджено Вченою радою Житомирського державного університету
імені Івана Франка протокол № 1 від 30.08.2013 р.*

Рецензенти:

Сейдаметова З.С. – доктор педагогічних наук, професор

Медведєв М.Г. – кандидат фізико-математичних наук, доцент

Вакалюк Т.А.

В14

Візуальне програмування. Навчально-методичний посібник для студентів фізико-математичного факультету. – Житомир: Вид-во ЖДУ, 2013. – 116 с.

Посібник призначений для використання студентами під керівництвом викладача на лекціях, практичних та лабораторних заняттях. Посібник містить лекційний курс та лабораторний практикум з програмування мовою Delphi. Викладений матеріал відповідає діючій програмі з Візуального програмування для спеціальності «Інформатика»

Для студентів фізико-математичних спеціальностей вищих педагогічних закладів, вчителів інформатики загальноосвітніх шкіл.

УДК 004.42+004.432.2

ББК 32.973.26-018.1

© Вакалюк Т.А., 2013

Зміст

Вступ

ЛЕКЦІЙНИЙ КУРС

Технологія візуального програмування.

Система візуального об'єктно-орієнтованого програмування Delphi.

Система візуального об'єктно-орієнтованого програмування Delphi.

Інтегроване середовище розробки (ICP) системи Delphi.

Система візуального програмування Delphi. Компоненти. Загальний огляд основних властивостей, методів та подій

Огляд бібліотеки візуальних компонентів (VCL) системи Delphi.

Компоненти введення та відображення текстової та чисельної інформації, дати та часу

Огляд бібліотеки візуальних компонентів (VCL) системи Delphi.

Компоненти відображення графічної та мультимедійної інформації.

Огляд бібліотеки візуальних компонентів (VCL) системи Delphi.

Компоненти - елементи керування.

Огляд бібліотеки візуальних компонентів (VCL) системи Delphi.

Компоненти – меню та компоненти-панелі.

Огляд бібліотеки візуальних компонентів (VCL) системи Delphi.

Компоненти – системні діалоги.

Повторне використання програмного коду. Загальний огляд механізмів збереження та повторного використання програмного коду. Шаблони та Депозитарії.

ЛАБОРАТОРНИЙ ПРАКТИКУМ

Загальні вимоги до виконання лабораторних робіт

Лабораторна робота №1.

Лабораторна робота №2

Приклад виконання практичного завдання.

Практичне завдання по варіантам.

Варіанти індивідуальних завдань:

Лабораторна робота №3

Приклад виконання практичного завдання.

Практичне завдання по варіантам.

Варіанти індивідуальних завдань:

Лабораторна робота №4

Приклад виконання практичного завдання.

Практичне завдання по варіантам.

Варіанти індивідуальних завдань:

Лабораторна робота №5

Практичне завдання по варіантам.

Варіанти індивідуальних завдань:

Лабораторна робота №6

Приклад виконання практичного завдання.

Практичне завдання по варіантам.

Варіанти індивідуальних завдань:

Лабораторна робота №7

Приклад виконання практичного завдання.

Практичне завдання по варіантам.

Варіанти індивідуальних завдань:

Лабораторна робота №8

Приклад виконання практичного завдання.

Практичне завдання по варіантам.

Варіанти індивідуальних завдань:

Лабораторна робота №9

Приклад виконання практичного завдання.

Практичне завдання по варіантам.

Варіанти індивідуальних завдань:

Лабораторна робота №10

Контрольні питання.

Практичне завдання по варіантам.

Варіанти індивідуальних завдань:

Лабораторна робота №11

Контрольні питання.

Практичне завдання по варіантам.

Варіанти індивідуальних завдань:

Лабораторна робота №12

Контрольні питання.

Практичне завдання по варіантам.

Варіанти індивідуальних завдань:

Лабораторна робота №13

Практичне завдання по варіантам.

Варіанти індивідуальних завдань:

Тестові завдання

Література

Вступ

Вивчення основ алгоритмізації та програмування є важливою складовою професійної підготовки майбутніх вчителів інформатики та ОТ, що дозволяє розкрити інтелектуальний потенціал майбутнього фахівця.

Курс "Візуальне програмування" є логічним продовженням у вивченні мов та систем програмування, але вже на якісно новому, сучасному рівні, з врахуванням тенденцій широкого застосування інформаційних технологій в навчальному процесі.

Проблемний характер лекції може бути реалізований за допомогою проблемно- символічних сигналів, що лежать в основі модельно-символічної технології організації розвивального навчання. Проблемна символіка може використовуватися як при викладанні нового матеріалу, так і при узагальненні матеріалу лекції. У першому випадку лектор записує на дошці чи то проектує проблемно-символічний сигнал (студенти отримують його запис разом із конспектом лекції), в основі якого лежить пара термінів (понять, принципів, положень), що характеризують тематику лекції. Подальше викладання матеріалу лекції супроводжується поступовим заповненням сигналу лектором. За такої організації роботи наприкінці лекції перед студентами буде "побудований" логічний конструкт теоретичного матеріалу із основними характеристиками предмету лекції. Слід зазначити, що можливий варіант, коли один із термінів опорної пари вже знайомий студентам. В такому випадку відбувається ознайомлення з новим матеріалом через призму вже відомого. На наступному етапі використання проблемної символіки можливе і при узагальненні матеріалу лекції. В такому випадку лектор, підсумовуючи сказане, пропонує аудиторії самостійно виконати проблемно-диференційоване завдання. Така робота на лекційному занятті сприятиме не просто репродуктивному засвоєнню нового програмного матеріалу, а й більш виразнішому, логічному зв'язку його з матеріалом попередніх лекцій, що в свою чергу позитивно впливає на рівень усвідомлення навчального матеріалу. Під час вивчення курсу передбачено обов'язкове виконання індивідуальних завдань. Завдання можуть бути як реферативного, так і практичного характеру (конкретні програмні реалізації).

ОБ'ЄКТ КУРСУ - технологія візуального програмування.

ПРЕДМЕТ - система візуального об'єктно-орієнтованого програмування Delphi.

МЕТА КУРСУ:

- ознайомлення із основними методологіями програмування та висвітлення процесу їх еволюції;
- загальний огляд основних технологій програмування;
- розгляд основних положень технології візуального програмування.

ЗАВДАННЯ КУРСУ:

- набути навичок практичної роботи із системами, що підтримують принципи візуального програмування;
- детально ознайомлення із системою візуального програмування Delphi.
- навчитися використовувати систему візуального програмування Delphi у власній професійній діяльності.

ЛЕКЦІЙНИЙ КУРС

Лекція №1

Технологія візуального програмування.

Мета: ознайомити студентів з поняттям технології візуального програмування.

Професійна спрямованість: дана лекція є складовою частиною професійної підготовки вчителя інформатики до викладання в школі.

Основні поняття: технологія, візуальне програмування, процедурне програмування.

План лекції:

1. Передумови виникнення та розвиток технології візуального об'єктно-орієнтованого програмування.

1.1. Основні поняття. Процедурне програмування.

1.2. Методології, що орієнтовані на дані (об'єкт). Об'єктно-орієнтоване програмування.

2. Основні положення та принципи технології візуального об'єктно-орієнтованого програмування.

2.1. Клас як новий тип даних.

Обрані методи: теоретичні, словесні, наочні, пояснювально-ілюстративний.

Наочність: схематичне зображення організації об'єкту, проблемно-символічні сигнали, що використовуються в ході викладання матеріалу.

Питання по темі для самостійного вивчення:

1. Історія створення та виникнення мови програмування Delphi.

2. Середовище програмування Delphi.

Запитання для самоаналізу та самоперевірки:

1. Що таке процедурне програмування?

2. Що таке візуальне програмування?

Рекомендована література: [1, 35-40;2,81-89;3,206-218;4,106-112]

Текст лекції.

1. Передумови виникнення та розвиток технології візуального об'єктно-орієнтованого програмування.

1.1. Основні поняття. Процедурне програмування.

Delphi – одна з найбільш потужних візуально об'єктно-орієнтованих систем програмування, що дозволяє на найсучаснішому рівні створювати як окремі прикладні програми, орієнтовані на роботу у Windows, так і розгалужені комплекси, що призначені для роботи в кооперативних мережах та в Інтернет.

Середовище візуального об'єктно-орієнтованого програмування Delphi – це графічна автоматизована оболонка над об'єктно-орієнтованою версією мови Pascal – Object Pascal.

Проект – це розроблюваний додаток, що включає сукупність файлів-модулів, файлів-форм та інші файли, які зв'язані власне файлом-проекту.

Тестові завдання

1. Об'єкт **Form** використовується для:
 - 1) Створення програмою нового вікна;
 - 2) Створення нової програми;
 - 3) Створення текстового файлу;
 - 4) Створення програмою нового файлу.
2. Властивість **Caption** об'єкта **Form** слугує для:
 - 1) Зміни заголовку форми;
 - 2) Зміни імені форми;
 - 3) Зміни підказки форми;
 - 4) Зміни імені графічного файлу;
3. Властивість **Icon** об'єкта **Form** слугує для:
 - 1) Зміни заголовку форми;
 - 2) Зміни імені форми;
 - 3) Зміни підказки форми;
 - 4) Задання піктограми, яка буде в заголовку форми під час виконання програми.
4. Об'єкт **Image** використовується для:
 - 1) Створення програмою нового вікна;
 - 2) Вставлення графічних файлів;
 - 3) Створення текстового файлу;
 - 4) Створення текстових полів у вікні програми.
5. Властивість **ReadOnly** об'єкта **Edit** - це:
 - 1) Можливість змінити текст (доступність поля);
 - 2) Текст у полі редагування;
 - 3) Текст підказки, яка висвітлюється, якщо навести курсор миші;
 - 4) Висвітлювати/не висвітлювати підказку.
6. Властивість **ShowHint** об'єкта **Edit** - це:
 - 1) Можливість змінити текст (доступність поля);
 - 2) Текст у полі редагування;
 - 3) Текст підказки, яка висвітлюється, якщо навести курсор миші;
 - 4) Висвітлювати/не висвітлювати підказку.
7. Об'єкт **PopupMenu** використовується для:
 - 1) Створення незалежного дво- чи трипозиційного прапорця;
 - 2) Створення головного меню;
 - 3) Створення контекстного меню;
 - 4) Створення багаторядкового редактора тексту.
8. Властивість **Lines** об'єкта **Memo** - це:
 - 1) Максимальна можлива кількість введених символів;
 - 2) Стан прапорця;
 - 3) Задання початкового тексту у полі редагування;
 - 4) Наявність смуг прокручування.
9. Властивість **State** об'єкта **CheckBox** - це:
 - 1) Максимальна можлива кількість введених символів;

- 2) Стан прапорця;
 - 3) Задання початкового тексту у полі редагування;
 - 4) Наявність смуг прокручування.
10. Властивість **Break** об'єктів **PopupMenu** та **MainMenu** - це:
- 1) Автоматичний виклик контекстного меню;
 - 2) Розбиття меню у горизонтальному напрямку;
 - 3) Команди меню;
 - 4) Комбінація «гарячих» клавіш для виклику команди меню.
11. Об'єкт **SpeedButton** призначений для:
- 1) Створення кнопки панелі інструментів;
 - 2) Створення головного меню;
 - 3) Створення контекстного меню;
 - 4) Створення у формі двовимірної таблиці символічних рядків.
12. Властивість **RowCount** об'єкта **StringGrid**- це:
- 1) Кількість рядків таблиці;
 - 2) Кількість стовпців таблиці;
 - 3) Кількість фіксованих стовпців таблиці, які не прокручуються зліва;
 - 4) Кількість фіксованих рядків таблиці, які не прокручуються вгору.
13. Властивість **DropDownCount** об'єкта **ComboBox**- це:
- 1) Кількість рядків у списку;
 - 2) Впорядкування за списком;
 - 3) Стиль оформлення і використання списку;
 - 4) Кількість рядків у випадяючому списку, які видимі без використання смуг прокручування
14. Delphi - це графічна автоматизована оболонка над версією мови:
- 1) Бейсік
 - 2) Паскаль
 - 3) Сі
15. Властивість **BorderStyle** об'єкта **Form** дає можливість:
- 1) Змінювати границі;
 - 2) Змінювати розміри вікна;
 - 3) Змінювати колір вікна;
 - 4) Змінювати колір границь.
16. Властивість **Color** об'єкта **Form** слугує для:
- 1) Зміни заголовку форми;
 - 2) Зміни імені форми;
 - 3) Зміни кольору форми;
 - 4) Зміни імені графічного файлу;
17. Властивість **Visible** об'єкта **Label** задає:
- 1) Видимість об'єкта;
 - 2) Доступність об'єкта;
 - 3) Вирівнювання тексту
18. Об'єкт **RadioButton** використовується для:
- 1) Введення користувачем рядка символів з клавіатури;
 - 2) Створення нової програми;
 - 3) Створення у формі засобу для використання однієї альтернативної можливості серед декількох;
 - 4) Створення текстових полів у вікні програми.

19. Які з перелічених властивостей є властивостями об'єкта **Form** (можливо декілька варіантів відповідей):

- | | | |
|-------------|-------------|------------|
| 1) Align; | 3) Icon; | 5) Name; |
| 2) Visible; | 4) Picture; | 6) Enabled |

20. Властивість **Name** об'єкта **Form** слугує для:

- | | |
|---------------------------|---------------------------------|
| 1. Зміни заголовку форми; | 3. Зміни підказки форми; |
| 2. Зміни імені форми; | 4. Зміни імені графічного файлу |

21. Об'єкт **Label** використовується для:

- 1) Створення програмою нового вікна;
- 2) Створення нової програми;
- 3) Створення текстового файлу;
- 4) Створення текстових полів у вікні програми.

22. Об'єкт **Edit** використовується для:

- 1) Введення користувачем рядка символів з клавіатури;
- 2) Створення нової програми;
- 3) Створення у формі засобу для використання однієї альтернативної можливості серед декількох;
- 4) Створення текстових полів у вікні програми

23. Властивість **Hint** об'єкта **Edit** - це:

- 1) Можливість змінити текст (доступність поля);
- 2) Текст у полі редагування;
- 3) Текст підказки, яка висвітлюється, якщо навести курсор миші;
- 4) Висвітлювати/не висвітлювати підказку.

24. Властивість **Checked** об'єкта **RadioButton** - це:

- 1) Можливість змінити текст (доступність поля);
- 2) Стан перемикача;
- 3) Текст підказки, яка висвітлюється, якщо навести курсор миші;
- 4) Висвітлювати/не висвітлювати підказку.

25. Об'єкт **MainMenu** використовується для:

- 1) Створення незалежного дво- чи трипозиційного прапорця;
- 2) Створення головного меню;
- 3) Створення контекстного меню;
- 4) Створення багаторядкового редактора тексту.

26. Властивість **ScrollBars** об'єкта **Memo** - це:

- 1) Максимальна можлива кількість введених символів;
- 2) Стан прапорця;
- 3) Задання початкового тексту у полі редагування;
- 4) Наявність смуг прокручування.

27. Властивість **Items** об'єкта **MainMenu** - це:

- 1) Автоматичний виклик контекстного меню;
- 2) Розбиття меню у горизонтальному напрямку;
- 3) Команди меню;
- 4) Комбінація «гарячих» клавіш для виклику команди меню.

28. Властивість **ShortCut** об'єктів **PopurMenu** та **MainMenu** - це:
- 1) Автоматичний виклик контекстного меню;
 - 2) Розбиття меню у горизонтальному напрямку;
 - 3) Команди меню;
 - 4) Комбінація «гарячих» клавіш для виклику команди меню.
29. Властивість **ColCount** об'єкта **StringGrid**- це:
- 1) Кількість рядків таблиці;
 - 2) Кількість стовпців таблиці;
 - 3) Кількість фіксованих стовпців таблиці, які не прокручуються зліва;
 - 4) Кількість фіксованих рядків таблиці, які не прокручуються вгору.
30. Властивість **FixedCols** об'єкта **StringGrid**- це:
- 1) Кількість рядків таблиці;
 - 2) Кількість стовпців таблиці;
 - 3) Кількість фіксованих стовпців таблиці, які не прокручуються зліва;
 - 4) Кількість фіксованих рядків таблиці, які не прокручуються вгору.
31. Властивість **Style** об'єкта **ComboBox**- це:
- 1) Кількість рядків у списку;
 - 2) Впорядкування за списком;
 - 3) Стиль оформлення і використання списку;
 - 4) Кількість рядків у випадяючому списку, які видимі без використання смуг прокручування
32. Об'єкт **Form** використовується для:
- 1) Створення програмою нового вікна;
 - 2) Створення нової програми;
 - 3) Створення текстового файлу;
 - 4) Створення програмою нового файлу.
33. Властивість **Caption** об'єкта **Form** слугує для:
- 1) Зміни заголовку форми;
 - 2) Зміни імені форми;
 - 3) Зміни підказки форми;
 - 4) Зміни імені графічного файлу;
34. Властивість **Icon** об'єкта **Form** слугує для:
- 1) Зміни заголовку форми;
 - 2) Зміни імені форми;
 - 3) Зміни підказки форми;
 - 4) Задання піктограми, яка буде в заголовку форми під час виконання програми.
35. Об'єкт **Image** використовується для:
- 1) Створення програмою нового вікна;
 - 2) Вставлення графічних файлів;
 - 3) Створення текстового файлу;
 - 4) Створення текстових полів у вікні програми.
36. Властивість **ReadOnly** об'єкта **Edit** - це:
- 1) Можливість змінити текст (доступність поля);

- 2) Текст у полі редагування;
 - 3) Текст підказки, яка висвітлюється, якщо навести курсор миші;
 - 4) Висвітлювати/не висвітлювати підказку.
37. Властивість **BorderStyle** об'єкта **Form** дає можливість:
1. Змінювати границі;
 2. Змінювати розміри вікна;
 3. Змінювати колір вікна;
 4. Змінювати колір границь.
38. Властивість **Color** об'єкта **Form** слугує для:
1. Зміни заголовку форми;
 2. Зміни імені форми;
 3. Зміни кольору форми;
 4. Зміни імені графічного файлу;
39. Властивість **Visible** об'єкта **Label** задає:
- 1) Видимість об'єкта;
 - 2) Доступність об'єкта;
 - 3) Вирівнювання тексту;
40. Об'єкт **RadioButton** використовується для:
1. Введення користувачем рядка символів з клавіатури;
 2. Створення нової програми;
 3. Створення у формі засобу для використання однієї альтернативної можливості серед декількох;
 4. Створення текстових полів у вікні програми.
41. Властивість **Text** об'єкта **Edit** - це:
1. Можливість змінити текст (доступність поля);
 2. Текст у полі редагування;
 3. Текст підказки, яка висвітлюється, якщо навести курсор миші;
 4. Висвітлювати/не висвітлювати підказку.
42. Об'єкт **Memo** використовується для:
1. Створення незалежного дво- чи трипозиційного прапорця;
 2. Створення головного меню;
 3. Створення контекстного меню;
 4. Створення багаторядкового редактора тексту.
43. Об'єкт **CheckBox** використовується для:
- 1) Створення незалежного дво- чи трипозиційного прапорця;
 - 2) Створення головного меню;
 - 3) Створення контекстного меню;
 - 4) Створення багаторядкового редактора тексту.
44. Властивість **MaxLength** об'єкта **Memo** - це:
- 1) Максимальна можлива кількість введених символів;
 - 2) Стан прапорця;
 - 3) Задання початкового тексту у полі редагування;
 - 4) Наявність смуг прокручування.
45. Властивість **AutoPopup** об'єкта **PopupMenu** - це:
- 1) Автоматичний виклик контекстного меню;
 - 2) Розбиття меню у горизонтальному напрямку;
 - 3) Команди меню;
 - 4) Комбінація «гарячих» клавіш для виклику команди меню.

46.Об'єкт **StringGrid** призначений для:

- 1) Створення кнопки панелі інструментів;
- 2) Створення головного меню;
- 3) Створення контекстного меню;
- 4) Створення у формі двовимірної таблиці символічних рядків.

47.Властивість **FixedRows** об'єкта **StringGrid**- це:

- 1) Кількість рядків таблиці;
- 2) Кількість стовпців таблиці;
- 3) Кількість фіксованих стовпців таблиці, які не прокручуються зліва;
- 4) Кількість фіксованих рядків таблиці, які не прокручуються вгору.

48.Об'єкт **ComboBox** призначений для:

- 1) Створення кнопки панелі інструментів;
- 2) Створення головного меню;
- 3) Створення випадаючого списку;
- 4) Створення у формі двовимірної таблиці символічних рядків.

49.Властивість **Sorted** об'єкта **ComboBox**- це:

- 1) Кількість рядків у списку;
- 2) Впорядкування за списком;
- 3) Стель оформлення і використання списку;
- 4) Кількість рядків у випадаючому списку, які видимі без використання смуг прокручування

Література

1. Архангельский А.Я. Программирование в Delphi 5.- М.:БИНОМ, 2000.- 1072 с.
2. Бобровский С. Delphi 5: учебный курс.- СПб.:Питер, 2001.-640 с.
3. Фаронов В.В. Delphi 6. Учебный курс. - М.:Издатель Молгачева С.В., 2001.-672 с.
4. Гофман В.Э., Хомоненко А.Д. Delphi 6. –СПб.:БХВ-Петербург, 2001.- 1152с.
5. Фаронов В.В., Шумаков П.В. Delphi 5. Руководство разработчика баз данных. - М.:Нолидж, 2000.-640 с.
6. Гофман В.Э., Хомоненко А.Д. Delphi 5. –СПб.:БХВ-Петербург, 1999.-800 с.
7. Гофман В.Э., Хомоненко А.Д. Работа с базами данных в Delphi. – СПб.:БХВ-Петербург, 2000.-656 с.
8. Прийма С.М. Методичні рекомендації до організації практичних робіт з інформатики в умовах впровадження модельно-символічної технології // Інформаційні технології в науці, освіті і техніці.- Черкаси:ЧДУ.-2002.- С.96-99.
9. Прийма С.М. Основні вимоги, подані до інтерфейсу користувача навчального програмного забезпечення // Тези Всеукраїнської науково-практичної конференції “Сучасний стан і перспективи шкільних курсів математики та інформатики у зв’язку з реформуванням у галузі освіти”.- Дрогобич:ДПУ.-2000.-С.167-170.
10. Прийма С.М. Про деякі аспекти використання систем візуального програмування в освіті // Теорія та методика навчання математики, фізики, інформатики.-Кривий Ріг:Видавничий відділ КДПУ.-2001.- С.137-142.
11. Прийма С.М., Єремєєв В.С. Використання системи проблемно-символічних сигналів при вивченні програмування // Інформаційні технології в освіті.-Бердянськ:БДП.-2001.-С.24-29.
12. Сван Т. Основы программирования в Delphi для Windows 95: Пер. с англ.- К.:Диалектика, 1996.-480 с.
13. Фаронов В.В. Delphi 5. Руководство программиста.-М.:Нолидж, 2001.-880 с.

Візуальне програмування

Навчально-методичне видання

ВАКАЛЮК Тетяна Анатоліївна

Візуальне програмування

*Навчально-методичний посібник для студентів
фізико-математичного факультету*

Надруковано з оригінал-макета автора

Підписано до друку 30.08.13. Формат 60x90/16. Папір офсетний.

Гарнітура Times New Roman. Друк різнографічний.

Ум. друк. арк. 6,8. Обл. вид. арк. 4. Наклад 300. Зам. 825.

Видавець і виготовлювач

Видавництво Житомирського державного університету імені Івана Франка

м. Житомир, вул. Велика Бердичівська, 40

Свідоцтво суб'єкта видавничої справи:

серія ЖТ №10 від 07.12.04 р.

електронна пошта (E-mail): zu@zu.edu.ua