

# ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ І АВТОМАТИЗАЦІЯ ПОЛІГРАФІЧНОГО ВИРОБНИЦТВА

УДК 004.4 232

*В.К. Овсяк<sup>1</sup>, О.В. Овсяк<sup>2</sup>, А.С. Василюк<sup>3</sup>.*

<sup>1</sup>Українська академія друкарства, *Politechnika Opolska*

<sup>2</sup>Львівська філія Київського національного університету культури і мистецтв

<sup>3</sup>Національний університет Львівська політехніка

## ЕЛЕМЕНТИ МОВИ ПРЕДМЕТНИХ УНІТЕРМІВ

*Описано створені знаки предметних унітермів, якими позначено об'єкти графічних інтерфейсів інформаційних систем.*

*The created symbols of objective uniterms are described, where the object of informational systems graphical interfaces are marked.*

### Вступ

В алгебрі алгоритмів розрізняються абстрактні і предметні унітерми [1 - 4]. Абстрактними унітермами є сталі та змінні, значення яких невідомі, або залежності, співвідношення, оператори, функції з невідомими зв'язками між складовими залежностей, співвідношень, операторів і функцій [4]. Предметні унітерми є конкретними сталими, змінними з відомими значеннями, а залежності, співвідношення, оператори і функції з відомими зв'язками між їхніми складовими [4]. Алгоритми утворені абстрактними унітермами, називаються абстрактними алгоритмами, а алгоритми, які отримані із абстрактних заміною абстрактних унітермів предметними, заданням значень сталих, змінних і залежностей, співвідношень, операторів та функції, називаються моделями абстрактних алгоритмів [4 - 7]. З моделей абстрактних алгоритмів можуть бути, при наявності відповідних комп'ютерних систем, згенеровані в автоматичному або, принаймні в автоматизованому, режимах коду комп'ютерних програм з використанням тих або інших алгоритмічних мов програмування. Наприклад, то може бути представлений найсучаснішими мовами програмування C# [8] та XAML [9]. Але для розв'язання задачі підвищення рівня автоматизації програмування генерування кодів програм з формул алгоритмів, перш за все необхідно створити граматику мови предметних унітермів. У даній праці подано і описано створені знаки предметних унітермів та синтаксис і семантику опису моделей графічних інтерфейсів інформаційних систем з використанням створених предметних унітермів.

### Таблиця знаків шрифту "Abstractal"

На рис.1 показана вікно знаків, які утворюють створений шрифт предметних унітермів. Шрифт предметних унітермів названо "Abstractal". Першим символом створеного шрифту є знак, який знаходиться у першому зверху рядку дев'ятого зліва стовпчика. Ним є знак хвильки (~), який використовується для задання назви графічних елементів інтерфейсу користувача інформаційних систем.

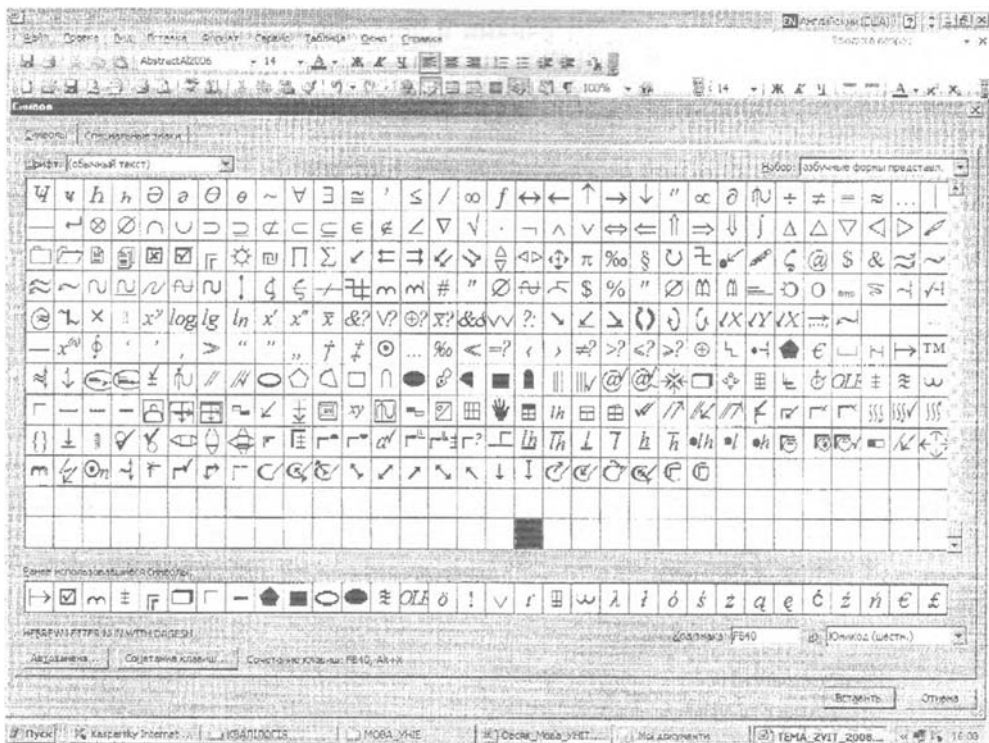


Рис. 1. Вікно знаків шрифту "Abstract".

### Умовні позначення предметних унітермів

З метою опису графічних об'єктів інтерфейсів к інформаційних систем та параметрів графічних об'єктів вводимо такі умовні позначення:  $\square$  - знак форми,  $\Gamma$  - панель,  $\square$  - кнопка,  $\square$  - знак альтернативного вибору,  $\odot$  - знак безальтернативного вибору,  $\sim$  - мітка,  $\swarrow$  - курсор,  $\Xi$  - головне меню,  $\Xi$  - підменю,  $\infty$  - закладка,  $\approx$  - текст бокс,  $l$  - довжина об'єкта,  $h$  - ширина об'єкта,  $lh$  - довжина і ширина,  $x$  - абсциса верхнього лівого рогу об'єкта,  $y$  - ордината верхнього лівого рогу,  $xu$  - координати верхнього лівого рогу об'єкта,  $\oplus$  - колір,  $\sim$  - текст,  $\updownarrow$  - шрифт і параметри шрифту,  $\updownarrow$  - перо і його параметри,  $\swarrow$  - цок лівою клав'яшею миші,  $\swarrow$  - подвійний цок лівою клав'яшею миші,  $\searrow$  - цок правою клав'яшею миші,  $\searrow$  - подвійний цок правою клав'яшею миші,  $\updownarrow$  - вертикальна прокрутка,  $\leftarrow$  - горизонтальна прокрутка,  $\updownarrow$  - горизонтально-вертикальна прокрутка,  $\Gamma$  - файл,  $\Gamma$  - запис у файл,  $\Gamma$  - зчитування з файлу,  $\leftarrow$  - ввід з клавіатури,  $\rightarrow$  - вивід на екран монітора,  $\oplus$  - відсвіжування об'єкта,  $-$  - пряма лінія,  $-$  - крива лінія,  $-$  - ширина лінії,  $\circ$  - еліпс,  $\bullet$  - зафарбований еліпс,  $\square$  - прямокутник,  $\blacksquare$  - зафарбований прямокутник,  $\triangle$  - багатокутник,  $\blacktriangle$  - зафарбований багатокутник.

### Синтаксис і семантика мови предметних унітермів

#### 1. Опис форми

Ідентифікатором форми є знак  $\Gamma$ , а параметри форми записуються у круглих відкритій і закритій дужках, розмішених справа від знаку форми. Напри-

клад, опис форми із заданими значеннями її довжини і ширини має такий вигляд

$$\Gamma(l = 700, h = 500),$$

де  $l = 700$  – піксельне значення довжини форми, а  $h = 500$  – піксельне значення ширини форми. Альтернативний, більш компактний, запис має такий вигляд

$$\Gamma(lh = 700\& 500),$$

де знак  $\&$  використовується як розділювач піксельних значень довжини і ширини.

Опис форми з розмірами 700 на 500 пікселів, кольором з назвою Cyan, наявністю на ній панелі ( $\square$ ) і кнопки ( $\square$ ), описується так

$$\Gamma(lh = 700\& 500, \text{c} = \text{Cyan}, \square, \square).$$

### 2. Опис панелі ( $\square$ )

Нехай для панелі потрібно задати координати лівого верхнього кута ( $xy$ ), розміри ( $lh$ ), колір ( $\text{c}$ ), наявність закладки ( $\text{m}$ ) і елементів альтернативного вибору ( $\square$ ). Унітерм з такими даними описується так

$$\square(xy = 10\&10, lh = 400\&300, \text{c} = \text{Yellow}, \text{m}, \square(57, 3, 10)),$$

де вираз  $\square(57, 3, 10)$  описує наявність елементів альтернативного вибору, назвами яких є номери 57, 3 і 10.

### 3. Кнопка ( $\square$ )

Допустимо, що для кнопки потрібно задати координати її лівого верхнього рогу ( $xy$ ), довжину і ширину ( $lh$ ) та назву алгоритму опрацювання цюку на кнопці лівою клавішею мишки ( $\text{c}$ ). Опис кнопки з цими даними є таким

$$\square(xy = 600\&20, lh = 30\&10, \text{c} = \text{RYS}()),$$

де  $\text{RYS}()$  – назва алгоритму опрацювання цюку на кнопці лівою клавішею мишки.

### 4. Опис елемента альтернативного вибору ( $\square$ )

Задання для елемента альтернативного вибору ( $\square$ ) його назви (57), координат розташування ( $xy$ ) і текстового позначення ( $\sim$ ) матиме такий вигляд

$$\downarrow 57(xy = 600\&100, \sim = \text{VybirAlter}_1).$$

Аналогічно описуються елементи альтернативного вибору з назвами – номерами 3 і 10.

$$\square 3(xy = 600\&120, \sim = \text{VybirAlter}_2).$$

$$\square 10(xy = 600\&140, \sim = \text{VybirAlter}_3).$$

### 5. Опис закладки (m)

Пехай для закладки (m) потрібно задати координати її лівого верхнього кута (xy), її назву (~), колір (⊗) і розміри (lh). Опис закладки з цими даними, наприклад, є таким

$m(xy = 300\&300, \sim = Zakladka, \otimes = Blue, lh = 250\&250).$

### Модель графічного інтерфейсу інформаційної системи

Модель графічного інтерфейсу інформаційної системи має включати всі потрібні елементи (форму, меню, панель, кнопки і т.д.). У зв'язку з цим, а також і тим, що для кожного елемента або задано місце його розташування, або ж місце розташування може бути задано автоматично, то модель графічного інтерфейсу описуємо з використанням операції секвентування. Так як всі без виключення графічні елементи розміщуються на формі, то унітерм форми має бути вложеним у всі операції секвентування. Аналогічно, якщо є графічні елементи, які розміщуються на панелі, то унітерм панелі має бути вложеним у ті всі секвенції в яких знаходяться унітерми, які розміщені на панелі. Так само мають бути описані всі решти графічні елементи, які містять графічні елементи.

Наприклад, модель графічного інтерфейсу, яка має описані вище унітерми матиме такий вигляд

$f(lh = 700\&500, \otimes = Yellow, \sim = \text{“ПАРП – С#”}, \square)$   
;  $f(xy = 10\&10, lh = 400\&300, \sim = \text{“panel1”}, \otimes = Orchid)$   
;  $\square(xy = 600\&20, lh = 30\&10, \sim = \text{“ДОДАЙ”}, \otimes = LightBlue)$   
;  $\square57(xy = 600\&100, \sim = VybirAlter_1)$  (1)  
;  $\square3(xy = 600\&120, \sim = VybirAlter_2)$   
;  $\square10(xy = 600\&140, \sim = VybirAlter_3)$

### Опис графічних елементів мовами C# і XAML

Одними з найсучасніших мов об'єктно - орієнтованого програмування є мови C# [8] і XAML [9].

Фрагмент програми C#, яким описується форма, панель і кнопка має такий вигляд:

```
using System.Windows.Forms; // 1

public class P_1: System.Windows.Forms.Form // 2

    Button prz1=new Button(); // 3
    Panel panell = new Panel(); // 4

    public P_1() // 5
    {
```

```

this.Width=700;           // 6
this.Height=500;         // 6.1
this.Name="ШАРП - С#"; // 7
this.BackColor=Color.Yellow; // 8

prz1.Location = new Point(600,20); // 9
    prz1.Width=30;           // 9.1
prz1.Height=10;           // 10
prz1.Text="ДОДАЙ";       // 10.1
prz1.BackColor=Color.LightBlue; // 11

panel1.Location = new Point(10,10); // 12
panel1.Name = "panel1"; // 13
panel1.Size = new System.Drawing.Size(400,300); // 14
panel1.BackColor=Color.Orchid; // 15

this.Controls.Add(prz1); // 16
this.Controls.Add(panel1); // 17

```

У першому рядку описано використання системного підпростору форм Forms. Другий рядок містить заголовка створюваного класу P\_1 і наслідування ним системного класу форми Form. Створення об'єктів системних класів кнопки (prz1) і панелі (panel1), описано у 3 і 4 рядках. Заголовок опису конструктора класу наведено у 5 рядку. Задання довжини, ширини, назви і кольору форми наведено у 6 – 8 рядках. Опис розташування лівого верхнього рогу, розмірів, назви і кольору кнопки описано у рядках 9 – 11. Для панелі, координати розташування її лівого верхнього рогу, назви, розмірів і кольору, наведено у 12 – 15 рядках. Опис знаходження кнопки і панелі на формі маємо у 16 і 17 рядках.

Порівнюючи наведений фрагмент коду і три перші рядки формули (1), якими описано тіж самі параметри графічних елементів інтерфейсу, очевидним є більш компактна форма запису отримувана з використанням введеного шрифту.

Нижче наведено фрагмент коду мовою XAML

```

<Window x:Class="TermEdit.MainForm"

xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"

Width="700"
Height="500"
Name="ШАРП - С#"
Background="Yellow" >

<Button Name="ДОДАЙ"
Width="20"
Height="28"
HorizontalAlignment="Left"

```

```
VerticalAlignment="Top">
</Button>

</Window>
```

У першому рядку описано клас форми, а другий і третій рядки містять дані фірми розробника і версію мови *XAML*. Наступні три рядки описують довжину, ширину, назву і колір фону форми. У подальших шести рядках задано назву, довжину, ширину кнопки та її прив'язку до лівої (`HorizontalAlignment "Left"`) і верхньої (`VerticalAlignment="Top"`) сторони форми.

Порівнюючи фрагмент опису мовою *XAML* з описом першого і третього рядка формули (1), які описують такі ж параметри як і фрагмент коду *XAML*, неважко оцінити що використання мови предметних унітермів забезпечує більш компактний опис.

### Висновки

1. Описано синтаксис і семантику елементів створеної мови предметних унітермів, яка включає систему спеціальних символів для однозначної ідентифікації, як параметрів так і самих графічних елементів інтерфейсів комп'ютерних систем.

2. Подано створений принцип побудови математичних моделей графічних інтерфейсів інформаційних систем, який полягає в описі операцією секвенування вложення графічних елементів та їхніх параметрів.

1. Овсяк В. Засоби еквівалентних перетворень алгоритмів. //Доповіді національної академії наук України. №9, 1996. – с.83-89.

2. Овсяк В. АЛГОРИТМИ: аналіз методів, алгебра впорядкувань, моделі, моделювання –Львів, 1996 -132с.

3. Овсяк В. АЛГОРИТМИ: методи побудови, оптимізації, дослідження вірогідності. – Львів "Світ", 2001. - 160 с.

4. Owskiak W., Owskiak A., Owskiak J. Teoria algorytmów abstrakcyjnych i modelowanie matematyczne systemów informacyjnych. - Opole: "Politechnika Opolska", 2005. -275s.

5. Бритковський В.М. Моделювання редактора формул секвенційних алгоритмів. Автореф. дис. к.т.н. – Львів: 2003. -18с.

6. Василюк А.С. Підвищення ефективності математичного і програмного забезпечення редактора формул алгоритмів. Автореф. дис. к.т.н. – Львів, 2008. -20с.

7. Овсяк О.В. Моделювання транслятора структур даних електромеханічних схем друкарських машин. Автореф. дис. к.т.н. – Львів, 2002. -18с.

8. Трозелсен Э. Язык программирования C. 2005 и платформа .NET 2.0. - Москва, Санкт-Петербург, Киев: "И.Д.Вильямс", 2007. -1168с.

9. Мак-Дональд М. WPF: Windows Presentation Foundation в NET 3.5 с примерами на C. 2008. Для профессионалов. - Москва, Санкт-Петербург, Киев: "И.Д.Вильямс", 2008. - 928 с.