

УДК 378:147:51:044.9

Вдовичин Т. Я., Лазурчак Л. В.

Дрогобицький державний педагогічний університет імені Івана Франка,  
Дрогобич, Україна

## ОСОБЛИВОСТІ ВИВЧЕННЯ ПРОГРАМУВАННЯ МАЙБУТНІМИ ВЧИТЕЛЯМИ ІНФОРМАТИКИ

DOI: 10.14308/ite000696

Процес підготовки майбутніх учителів інформатики передбачає формування знань і вмінь, а також професійних навичок у контексті отримання майбутньої кваліфікації «вчитель інформатики». Вивчення програмування дає базові фундаментальні поняття та способи їх подальшого застосування в різних дисциплінах, підтримує міжпредметні зв'язки протягом усього процесу навчання студентів та в подальшій професійній діяльності. Ця стаття присвячена висвітленню особливостей навчання програмування засобами середовища C++ Builder для майбутніх учителів інформатики.

C++ Builder – одна з найпотужніших систем, що дозволяє на найсучаснішому рівні створювати як окремі прикладні програми Windows, так і розгалужені комплекси, призначені для роботи в корпоративних мережах і в Інтернет. Це інструмент для швидкої розробки додатків на C++ під Windows, який підтримує можливість програмування, що ґрунтується на компонентах. У статті досліджено можливості цієї системи саме для розробки складних і ефективних прикладних програм для майбутніх учителів інформатики. Акцентовано на візуальній розробці інтерфейсу програмного забезпечення, що дозволяє спростити взаємодію програмного продукту з користувачем та додати користувачькому інтерфейсу якості професійного рівня. Результатом візуального проектування є скелет майбутньої програми, у яку вже внесені відповідні коди. Інструменти візуальної розробки забезпечують продуктивність багаторазового використання візуальних компонентів у поєднанні з удосконаленими засобами доступу до баз даних.

У статті досліджено візуальні можливості мови програмування C++ для майбутніх учителів інформатики на прикладі створення калькулятора та текстового редактора. Простота освоєння середовища C++ Builder та підтримки широкого спектра технологій робить його універсальним інструментом створення програмних проектів найрізноманітнішої спрямованості. Засоби середовища C++ Builder дозволяють майбутнім учителям інформатики створювати різноманітні навчальні програми, що можуть бути використані у професійній діяльності, зокрема у школі.

**Ключові слова:** майбутні вчителі інформатики, мова програмування, C++ Builder, об'єктно-орієнтоване програмування, візуалізація, візуальне програмне забезпечення.

**Постановка проблеми.** Сьогодні відбувається стрімкий розвиток процесу інформатизації суспільства. Інформаційні технології проникають практично у всі сфери соціальної практики і стають невід'ємною частиною життя людини. Для роботи з великими обсягами інформації все частіше використовують персональний комп'ютер, зокрема виникає потреба в розробці програмних продуктів.

Підготовка студентів зорієнтована на розвиток здатності сприймати нові знання, спонукання до педагогічних пошуків, виявлення власної активності та вміння реалізовувати набуті знання не тільки під час навчання у ЗВО, а й у подальшій діяльності. Зміни, що постійно відбуваються у суспільстві, спонукають до професійного самовизначення та уміння



жити в динамічному середовищі, до вміння визначати мету діяльності, прогнозувати варіанти її досягнення. Найбільш актуальною ця проблема постає для студентів педагогічних університетів, тобто майбутніх педагогів, оскільки система освіти постійно реформується для учнів, батьків і вчителів.

Аналізуючи процес навчання майбутніх учителів інформатики, важливим є засвоєння студентами фундаментальних понять, набуття навичок практичної роботи з програмними засобами. Основний шлях реалізації завдання – навчити студента методологічному мисленню, надати відомості з предметної галузі, сформувати вміння до практичного застосування, втілювати набуті навички у практичну діяльність.

Підготовка фахівців першого (бакалаврського) рівня вищої освіти галузі знань 01 «Освіта» спеціальності 014 «Середня освіта (інформатика)» передбачає знайомство з нормативними та варіативними дисциплінами. Що стосується фахових дисциплін, то саме на першому році навчання викладається дисципліна «Програмування». Під час вивчення цієї дисципліни студенти повинні сформувати базу знань, вмінь та отримати навички, що їх будуть супроводжувати і при вивченні наступних дисциплін, а також у подальшій професійній діяльності.

Предметом навчальної дисципліни є теорія і практика застосування у програмуванні базових алгоритмічних структур і даних, що ґрунтується на сучасних технологіях розробки програмного забезпечення. Основними завданнями є: вивчення принципів процедурного та об'єктно-орієнтованого програмування щодо розробки програм; вивчення типових підходів до аналізу найбільш розповсюджених алгоритмів вирішення прикладних задач; вироблення вміння реалізації типових алгоритмів в різних середовищах програмування; поглиблення уявлення про застосування інформаційних технологій в прикладній інформатиці.

У результаті вивчення навчальної дисципліни майбутньому вчителю інформатики пропонується ознайомитися з системою C++ Builder. Це надзвичайно потужна система розробки прикладних програм для Windows. Вона впевнено займає передові позиції як серед професійних програмістів, так і серед людей, що ніколи професійно не займалися програмуванням. Перевагами C++ Builder є можливості використання мови C++, що лежить в його основі. Це одна з найпотужніших сучасних алгоритмічних мов загального призначення. Разом зі своєю простотою, C++ Builder дає широкі можливості для розробки складних і ефективних програм, забезпечує швидкість візуальної розробки та продуктивність, володіє удосконаленими інструментами та різномасштабними засобами доступу до баз даних. C++ Builder може бути використаний там, де потрібно доповнити існуючі додатки розширеним стандартом мови C++, підвищити швидкодію та додати користувальницькому інтерфейсу якості професійного рівня.

Під час навчання програмування пропонується сформувати у майбутніх учителів інформатики компетентність, що необхідна для успішного виконання поставлених завдань. Компетентнісний підхід полягає в набутті та розвитку у студентів набору ключових та предметних компетентностей, що визначають його успішну адаптацію в суспільстві, здатність вирішувати життєві й професійні проблеми, а також включають вузько професійні знання та уміння, що характеризують кваліфікацію майбутнього вчителя інформатики. Важливими є такі якості, як ініціативність, співпраця, здатність до роботи в колективі, комунікативні здібності.

**Аналіз останніх досліджень та публікацій.** Короткий огляд основних базових конструкцій мови C++ та їх практичне застосування висвітлено у працях Т. Павловської, Ю. Щупак [11], С. Загребельного [9], Т. Сопронюка [12]. Методика опанування основами мови C++ і створення проектів у C++ Builder апробована Я. Глинським, В. Анохіним, В. Ряжською [14], Л. Лазурчак, Т. Вдовичин [6].

Актуальними є дослідження особливостей розроблення об'єктно-орієнтованих програм мовою C++ у роботах Ю. Грицюк [13], Т. Рак, Г. Шілдрт [3], В. Шаміс, М. Вайсфельд [2], С. Гилберт [7], Б. Маккарти, Д. Холингворт, Б. Сворт, М. Кешмен, П. Густавсон [4].

Проблемою систематизації базових засобів процедурного програмування мовою С++ займалися І. Дудзяний [5], С. Глушаков, Н. Дейл, Д. Якушев. Проектування і створення програмного забезпечення з допомогою С++ детально проаналізовано у книзі Б. Страуструпа [1], що є повною довідковою системою для вивчення мови програмування С++.

**Актуальність дослідження.** Програмування є базовою дисципліною у підготовці майбутніх учителів інформатики, а набуті знання та вміння будуть супроводжувати студентів протягом всього навчання у ЗВО. Практичні навички стануть у нагоді при вивченні інших навчальних дисциплін, а також проходженні педагогічної практики, підготовці курсових проектів та кваліфікаційних робіт.

У результаті вивчення програмування у майбутніх вчителів інформатики повинна бути сформована предметна компетентність, а саме щодо реалізації типових алгоритмічних конструкцій у середовищі С++ Builder; здійсненні супроводу програм та внесення змін за допомогою вбудованих інструментальних засобів; розроблення проектів з використанням парадигм об'єктно-орієнтованого програмування.

Під час вивчення програмування майбутнім учителям інформатики подаються відомості про різні середовища програмування, їх порівняльна характеристика, а також обґрунтування вибору системи С++ Builder. Зокрема, Microsoft Visual Studio – це середовище розробки додатків під ОС Windows. MS Visual Studio надає користувачеві при створенні проекту велику кількість різних типів проектів, що, безумовно, плутає навіть бувалого програміста, не те що початківця. Але велика кількість типів проектів дає гарне уявлення про можливості цього інструменту. Редактор цього середовища підсвічує синтаксис і виконує форматування коду, автоматично завершує деякі структури коду. Це потужний інструмент, що ідеально підходить для розробки великих проектів, однак для початківця є достатньо громіздким.

Тому, у зв'язку з комерціалізацією програмних продуктів, актуальність дослідження не викликає сумніву. Через недостатність державних коштів, купівля ліцензійних офісів відходить на другий план. Виникає необхідність створення власних програмних продуктів, що працюватимуть не гірше стандартних програмних додатків. За допомогою мови програмування С++ Builder створення таких додатків є легким та надійним.

Аналізуючи процес підготовки майбутніх учителів інформатики, слід дослідити вивчення дисципліни «Програмування» під час аудиторних занять та самостійної роботи студентів. Як видно з рис. 1, частка аудиторних годин та годин, що виносяться на самостійне опрацювання, розділена як 36% до 64%.

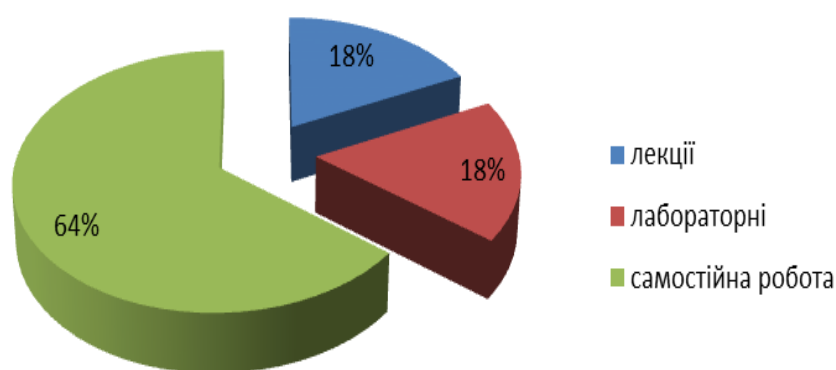


Рис. 1. Розподіл навчальних годин з дисципліни «Програмування»

Тому для обґрунтування змісту дисципліни «Програмування» для майбутніх учителів інформатики підібрано теоретичний та практичний матеріал, що є доцільним для застосування у процесі навчання; розроблено рекомендації щодо виконання лабораторних

робіт та до тем, які виносяться на самостійне опрацювання. Потужним стимулом для вивчення студентами програмування є можливість вивчати і закріплювати навчальний матеріал з різних дисциплін, що дають змогу працювати у власному темпі навчальної діяльності, самостійно планувати хід навчання з урахуванням пропозицій викладача і власного досвіду (рівня самостійної пізнавальної діяльності, загальних здібностей та потреб) тощо.

Тому **метою дослідження** є обґрунтування особливостей вивчення програмування для навчання майбутніх учителів інформатики.

**Предмет дослідження** – створення програмних додатків у середовищі C++ Builder.

Для досягнення мети і реалізації завдань дослідження застосовувався комплекс методів: аналіз досвіду щодо використання можливостей C++ Builder; спостереження – для виявлення особливостей побудови користувацьких інтерфейсів з використання інтегрованого середовища C++Builder; самооцінювання – для дослідження особистісних умінь майбутніх учителів інформатики створювати та використовувати програмні додатки за допомогою мови програмування C++.

**Виклад основного матеріалу.** Важливим аспектом процесу навчання майбутніх учителів інформатики з програмування є організація роботи на аудиторних заняттях та самостійної роботи, а саме: спочатку результатом роботи є формування знань, що дають змогу вирішувати типові завдання; згодом студенти виконують складніші завдання, що потребують розвитку практичних навичок з розв'язування навчальних задач програмування, у процесі виконання яких студенти аналізують отримані результати, роблять відповідні висновки, завдяки чому у них формується критичне мислення, підвищується їхній компетентісний рівень; самостійна робота студентів, коли завдання передбачають пошук і реалізацію його виконання, що виходить за межі відомих студенту знань і методів вирішення тощо.

Широкий набір можливостей розв'язування прикладних задач у середовищі C++ Builder дають змогу для застосування у професійній діяльності майбутніх вчителів інформатики. Варто зосередити увагу на завданнях професійно-практичного профілю, що можна запропонувати студентам для вивчення як під час аудиторних занять, так і для самостійного опрацювання.

У C++ Builder можлива віртуальна реалізація різноманітних сучасних користувацьких інтерфейсів, що вдало використовуються на практиці. Розглянемо деякі з них.

Одна з популярних програм, що широко поширена у побуті, – калькулятор. Існують різні види калькуляторів з певним призначенням: інженерний, економічний, науковий тощо. Можна створити звичайний консольний калькулятор засобами C++ Builder, що виконуватиме найпростіші арифметичні операції (додавання, віднімання, множення, ділення), обчислюватиме тригонометричні функції тощо.

Для **створення власного калькулятора** спочатку слід ознайомитися із компонентом BitBtn (рис. 2), що розташований на вкладці *Additional Палітри компонентів* та детально проаналізувати його властивості.



Рис. 2. Компонент BitBtn

Це керуюча кнопка, що має можливість відобразити на своїй поверхні певне растрове зображення. Розглянемо основні властивості компонента.

Властивість Glyph цього компонента відповідає за зображення на кнопці. Glyph – це зображення, що, як правило, має формат растрового файлу Windows (\*.bmp) та складається з одного або більше растрових зображень, які відповідають чотирьом можливим станам

кнопки. Щоб визначити Glyph для BitBtn, потрібно двічі клікнути в *Інспекторі об'єктів* у полі, розміщеному напроти властивості Glyph цього компонента. У вікні, що відкрилося, за допомогою кнопки Load слід перейти в звичайне вікно відкриття файла рисунка і вибрати файл, що містить потрібне зображення. C++Builder пропонує велику кількість зображень для кнопок.

Властивість Kind визначає тип кнопки. По замовчуванню – значення властивості bkCustom, тобто користувач визначає сам Glyph і інші властивості кнопки. Можна встановити і типи: bkOK, bkCancel, bkHelp, bkYes, bkNo, bkClose, bkAbort, bkRetry, bkIgnore, bkAll. Для цих типів вже заготовлені відповідні надписи, введені піктограми, задані ще деякі властивості.

Ще один компонент, що буде використаний для створення калькулятора, Edit – однорядковий елемент редагування. Він розташований на вкладці Standard *Палітри компонентів* і призначений для введення тексту. Основна його властивість – Text – дозволяє читати і записувати деякий текст в поле Edit. Інформація, записана в поле Edit, є стрічкового типу. А отже, якщо користувач вводить в Edit деяке число, то слід виконати додаткове перетворення, наприклад : ... int z = StrToInt (Edit1->Text)... Крім функції StrToInt є обернена до неї IntToStr, що переводить ціле число в стрічку. Функції StrToFloat і FloatToStr переводять стрічку в дійсне число і дійсне число в стрічку відповідно.

Для того щоб створити калькулятор, на форму ( Form1) потрібно розмістити 26 кнопок (компонент BitBtn) та зробити напис на кожній із них, виконавши послідовно наступні дії:

- 1) натиснути на потрібну кнопку, тобто зробити її активною;
- 2) у вікні Object Inspector на вкладці Properties задати властивість Caption для поміченої кнопки (наприклад «1»);

Для візуалізації результату потрібно вибрати компонент Edit на вкладці Standard *Палітри компонентів*. За допомогою мишки слід розмістити Edit у верхній частині робочої області вікна форми (рис. 3).

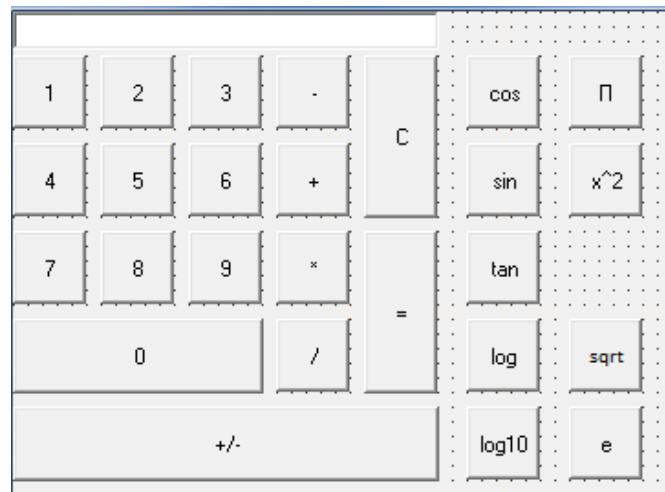


Рис. 3. Загальний вигляд калькулятора

Для того щоб запрограмувати кнопку на необхідну функцію, слід виконати подвійний клік на ній. Відбувається автоматичний перехід у вікно Unit1, де прописуються необхідні функції. Розпочинати слід з кнопок 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 (рис. 4). Оголошення змінних відбувається на початку програмного коду:

```
int a, b, c;
char k;
```

```
void __fastcall TForm1::BitBtn16Click(TObject *Sender)
{
    Edit1->Text=Edit1->Text+"0";
    b=StrToInt (Edit1->Text);
}
```

Рис. 4. Програмний код кнопки «0»

*Коментар до коду:* { записує в віконці Edit1 нуль, переводить текст з віконця Edit1 в число і запам'ятовує його в змінній b }

За аналогією програмується кожна наступна цифра.

Далі програмуються кнопки «+» (додавання), «-» (віднімання), «\*» (множення), «/» (ділення). Програмний код для кнопки «+» зображено на рис. 5.

```
void __fastcall TForm1::BitBtn8Click(TObject *Sender)
{
    a=StrToInt (Edit1->Text);
    k='+';
    Edit1->Text="";
}
```

Рис. 5. Програмний код кнопки «+» (додавання)

*Коментар до коду:* { переводить текст з віконця Edit1 в число і запам'ятовує його в змінній a; символній змінній k присвоюється значення «+»; очищає віконці Edit1. Аналогічно для кожної наступної операції «-», «\*», «/» }

Щоб запрограмувати кнопку «=» (результат), слід використати оператор вибору switch, а саме:

```
switch(k)
{
    case '+': a= a+b;
                Edit1->Text=IntToStr(a); break;
    case '-': a= a-b;
                Edit1->Text=IntToStr(a); break;
    case '*': a= a*b;
                Edit1->Text=IntToStr(a); break;
    case '/': a= a/b;
                Edit1->Text=IntToStr(a); break;
}
```

*Коментар до коду:*

- якщо змінна k має значення «+», то змінна a присвоює значення «a + b», конвертується в рядок і виводиться на екран у віконці Edit1;
- якщо змінна k має значення «-», то змінна a привласнює значення «a – b», конвертується в рядок і виводиться на екран у віконці Edit1;
- якщо змінна k має значення «\*», то змінна a привласнює значення «a \* b», конвертується в рядок і виводиться на екран у віконці Edit1;
- якщо змінна k має значення «/», то змінна a привласнює значення «a / b», конвертується в рядок і виводиться на екран у віконці Edit1.

Можна продемонструвати програмні коди інших кнопок цього калькулятора (таблиця 1).

Таблиця №1.

*Демонстрація фрагментів програмних кодів для кнопок калькулятора*

Кнопка	Програмний код
«C» (очищення)	Edit1->Text="";

Кнопка	Програмний код
зміна знаку «+/-»	Edit1->Text=IntToStr(-b); b=- b;
«cos»(косинус)	Edit1->Text= FloatToStr(cos(StrToFloat(Edit1->Text)));
«sin»(синус)	Edit1->Text= FloatToStr(sin(StrToFloat(Edit1->Text)));
«log»(логарифм)	Edit1->Text= FloatToStr(log(StrToFloat(Edit1->Text)));
«e»(стала)	Edit1->Text= “2,71”;
«x^2» (число піднесене до квадрату)	Edit1->Text= pow(StrToFloat(Edit->Text), 2);
«sqrt» (корінь квадратний)	if (StrToFloat(Edit1->Text) >=0 Edit1-> Text = FloatToStr(sqrt(StrToFloat(Edit1->Text))); Else Edit1->Text = “Немає коренів!”;

Як результат, засобами C++ Builder створено найпростіший звичайний консольний калькулятор, що виконує найпростіші арифметичні операції, обчислює тригонометричні функції тощо. Ознайомившись із запропонованими у статті фрагментами програмних кодів, за аналогією можна додати до створеного калькулятора інші функції, що знаходяться в бібліотеці math.h.

Засобами C++ Builder можна створити **найпростіший текстовий редактор** з іменем Edit, що дозволить створювати нові текстові файли, відкривати вже існуючі на диску, редагувати та зберігати їх. Слід передбачити можливість роботи з буфером обміну та панеллю інструментів з кнопками, що забезпечить швидкий доступ до пунктів меню.

Розпочинаючи роботу з текстовим редактором, спочатку слід ознайомитися із компонентами Memo, MainMenu, PopupMenu вкладки *Standard Палітри компонентів*, компонентами Open Dialog, Save Dialog, Font Dialog з вкладки Dialogs, BitBtn з вкладки Additional, детально проаналізувати їх властивості.

Текстовий редактор призначений для роботи з текстовими рядками, тому на форму компонент Memo потрібно помістити вікно редагування багаторядкового тексту.

Компонент MainMenu дозволить створити головне меню текстового редактора. На формі слід помістити компоненти Panel – майбутня інструментальна панель редактора, що міститиме три стандартні кнопки BitBtn з вкладки Additional. Вони забезпечать відкриття та збереження файлів, встановлення потрібних шрифтів.

При роботі з файлами редактор повинен звертатися до дискових накопичувачів, а тому слід додати на форму діалоги Open Dialog і Save Dialog. Полегшить налаштування шрифтів компонент Font Dialog. Ці діалоги є невидимими на етапі виконання програми, а тому місце їх розміщення на формі немає значення.

Останнім компонентом форми буде PopupMenu (контекстне меню), що з'являється при натисканні правої кнопки миші і прив'язане до конкретного компонента. Формування контекстного меню виконується за допомогою Конструктора Меню, що викликається подвійним кліком на PopupMenu.

Налаштування властивостей попередньо розглянутих компонентів потрібно розпочати з головного компонента – форми програми. Спочатку слід змінити властивість Caption цього об'єкта на назву програми редактора – Текстовий редактор. Властивості Position встановити poDesigned, що забезпечить центрування створюваного редактора.

Далі слід попрацювати з компонентом Memo. Змінити властивість Lines, натиснувши кнопку з трьома крапками правіше Tstrings цієї властивості. У вікні редактора рядка String List Editor необхідно видалити рядок Memo1. Додати компоненту Memo для лінійки прокрутки потрібно для забезпечення можливості перегляду і редагування великих зон тексту за допомогою миші. Для властивості ScrollBars необхідно вибрати значення ssBoth.

Для того щоб працювало контекстне меню, необхідно у властивості PopupMenu вибрати значення PopupMenu1, яке стане доступним після переміщення на форму проекту діалогу PopupMenu.

Наступним кроком створення текстового редактора є розміщення на формі панелі інструментів `Panel1`. На ній будуть розміщені три кнопки. Щоб налаштувати властивості кнопок відповідно до виконуваних ними функцій, слід змінити по черзі властивість `Caption` для всіх трьох кнопок на Відкрити, Зберегти, Шрифт відповідно. Для того щоб тип курсора змінювався при попаданні на ці кнопки, потрібно змінити ще одну групову властивість кнопок. Для цього всі три кнопки слід виділити і в інспекторі об'єктів змінити властивість `Cursor` на значення `crHandPoint`.

Налаштування головного меню слід зробити за допомогою дизайнера меню, що викликається подвійним клацанням лівої кнопки миші по компоненту `MainMenu`. По горизонталі пункти меню потрібно ввести так: Файл, Формат, Правка, Допомога, Панель інструментів.

Тепер необхідно заповнити пункти меню по вертикалі. Вибрати пункт меню Файл і перейти за допомогою миші на нижче розміщене порожнє поле. Властивість `Caption` цього пункту меню слід змінити на Створити. Наступні пункти слід назвати: Відкрити, Зберегти, Зберегти як, Вихід. Якщо перед довільною буквою назви команди вставити символ `&` (амперсанд), то наступна за цим символом буква буде підкреслена. Цей спосіб виділення букви використовується для виконання команди по гарячій клавіші.

Аналогічно слід розширити по вертикалі пункт меню Формат, додавши пункт Шрифт. Пункт меню Правка наповнити пунктами Вирізати, Копіювати, Вставити. Пункт меню Допомога повинен видати повідомлення про допомогу. При активації пункту Панель інструментів виводиться на екран або приховується.

Тепер треба налаштувати компонент `FontDialog1`. Основною властивістю цього компонента є `Font`, за допомогою якого можна вибрати тип шрифту, його колір, розмір, стиль і т.п. (рис. 6). Ці установки будуть використовуватися в програмі редактора за замовчуванням.

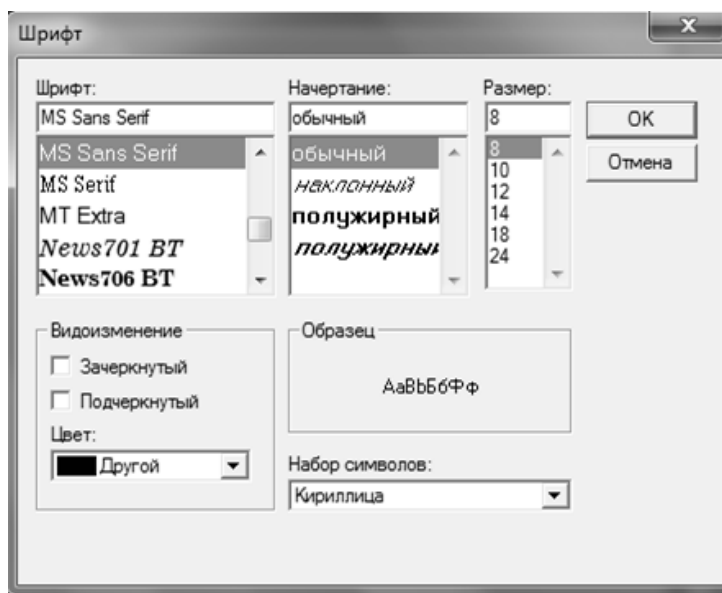


Рис. 6. Налаштування шрифту

Налаштувати контекстне меню `PopupMenu` можна аналогічно до головного меню `MainMenu`, додавши пункти з назвами Шрифт, Очистити, Вирізати, Копіювати, Вставити через властивість `Caption` (рис. 7).



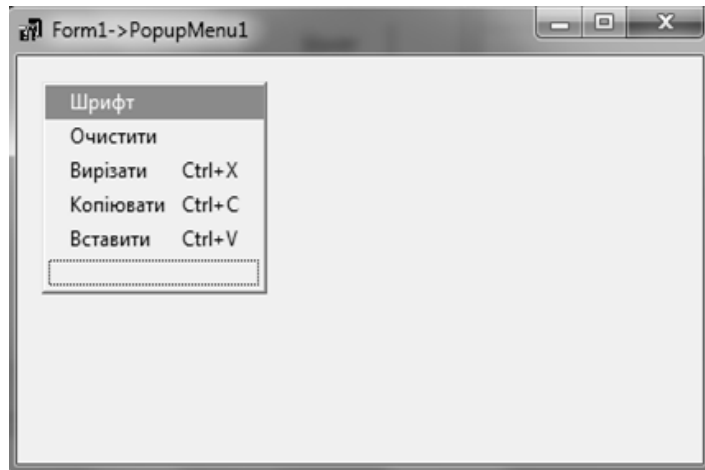


Рис. 7. Налаштування контекстного меню

Властивість PopupMenu головної форми додатка отримує назву PopupMenu1.

Налаштування компонентів завершено й отримана візуальна оболонка програми редактора (рис. 8).

Тепер слід запрограмувати кнопки та команди, натиснувши двічі на кнопку або відповідно на команду (таблиця 2).

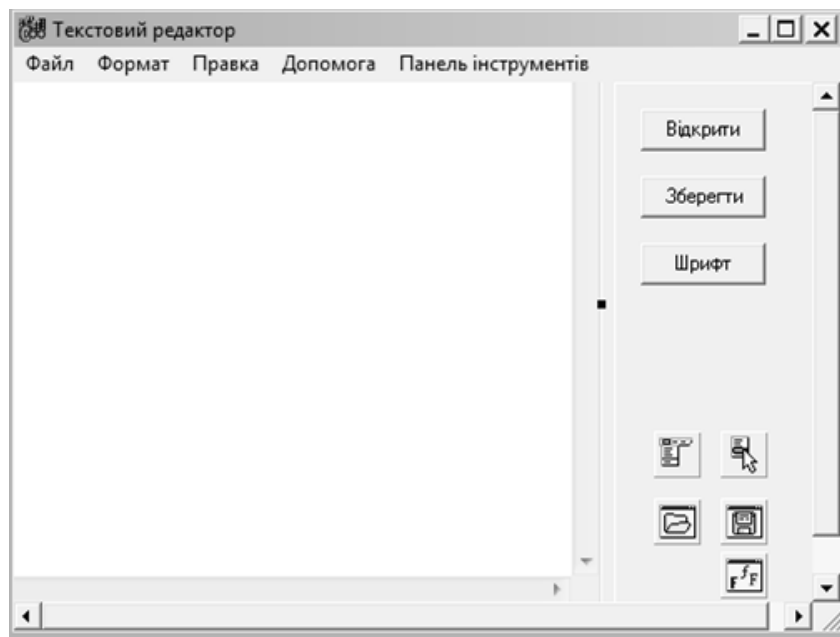


Рис. 8. Загальний вигляд форми

Таблиця № 2.

*Демонстрація програмних кодів для кнопок та команд текстового редактора*

Кнопка або команда	Програмний код
функція для команди головного меню Відкрити	<code>if (OpenDialog1-&gt;Execute( )) Memo1-&gt;Lines-&gt;LoadFromFile(OpenDialog1-&gt;FileName);</code>
функція для команди збереження файлів Зберегти	<code>if (SaveDialog1-&gt;Execute( )) Memo1-&gt;Lines-&gt;SaveToFile(SaveDialog1-&gt;FileName);</code>
функція для команди закриття файлів Закрити	<code>Close ( );</code>
команда Шрифт з пункту Формат головного меню	<code>if (FontDialog1-&gt;Execute( )) Memo1-&gt;Font=FontDialog1-&gt;Font;</code>

Кнопка або команда	Програмний код
команда Вирізати з пункту Правка головного меню	Memo1->CutToClipboard ();
команда Копіювати з пункту Правка головного меню	Memo1->CopyToClipboard ();
команда Вставити з пункту Правка головного меню	Memo1->PasteFromClipboard ();
команда Допомога	ShowMessage(“Для отримання додаткових відомостей щодо вашого запитання зверніться в сервіс підтримки Windows або шукайте відповідь в інтернет ресурсі”);
команда Панель інструментів	N9->Checked=!N9->Checked; Panel1->Visible=N9->Checked;

Створення програми редактора закінчено, можна приступити до його перевірки. Як результат, засобами C++ Builder створено найпростіший текстовий редактор, що дозволяє створювати нові текстові файли, відкривати вже існуючі на диску, редагувати та зберігати їх.

Для майбутніх учителів інформатики представлено завдання професійно-практичного спрямування у середовищі C++ Builder. Зокрема, акцентовано увагу на візуалізації як графічного відображення складних процесів на екрані комп'ютера у вигляді графічних примітивів. Візуалізуються будь-які процеси: управління, побудови, малювання тощо. Візуалізувати можна інтерфейси програмного забезпечення. Саме наявність візуальних засобів побудови інтерфейсів для Windows в C++ Builder, а також створюване ними візуальне програмне забезпечення, закріпили за ним термін «візуальне програмування».

**Висновки.** Структура навчальної дисципліни «Програмування», рекомендацій до виконання лабораторних занять і самостійної роботи сприятиме успішній реалізації всіх поставлених завдань щодо оволодіння майбутніми вчителями інформатики практичних навичок з розв'язування задач програмування. Вивчення основних понять, принципів розробки програм, основних структур даних, інформаційних моделей для задач програмування, проектування, розробка та налагодження програм є потенційною базою для здобуття студентами кваліфікації «вчитель інформатики». Закладені фундаментальні знання з програмування слугуватимуть надійним підґрунтям у процесі навчання, а оволодіння елементами технології створення програм у обраному середовищі програмування – базових знань, умінь та навичок не тільки під час навчання у ЗВО, а й у майбутній професії.

C++ Builder – одна з найпотужніших систем, що дозволяє на найсучаснішому рівні створювати як окремі прикладні програми Windows, так і розгалужені комплекси, призначені для роботи в корпоративних мережах і в Інтернеті. Система розробки прикладних програм C++ Builder актуальна як серед студентів, так і серед професійних програмістів та людей, що ніколи професійно не займались програмуванням. Використання мови C++, як однієї з найпотужніших сучасних алгоритмічних мов загального призначення, є базовою для C++ Builder, що володіє широкими можливостями для розробки складних і ефективних програм, забезпечує швидкість візуальної розробки та продуктивність.

У дослідженні були проаналізовані питання відносно побудови програмних додатків з простим інтерфейсом користувача в середовищі C++ Builder.

Створений програмний додаток калькулятор допоможе виконати різні математичні операції, є простим у користуванні та дає достовірні результати. У цьому дослідженні розглянуто процес створення найпростішого калькулятора з обмеженою кількістю запропонованих операцій для проведення обчислень. Калькулятор працює з цілими числами. У подальших дослідженнях варто розглянути процес створення та налаштування калькулятора для опрацювання дійсних чисел та виконання складніших математичних операцій.

Текстовий редактор – програма для створення та редагування текстових документів. Дані програми є незмінними у навчанні та роботі і легко реалізуються у даному середовищі. У статті використано можливості роботи з діалоговими вікнами, набору та редагування тексту, застосовуючи контексне меню та буфер обміну. В подальших дослідженнях варто проєкспериментувати з можливостями середовища C++ Builder, зокрема більш детально ознайомитися з компонентами бібліотеки візуальних компонентів (VCL).

Отже, поєднання простоти освоєння візуального середовища та підтримки широкого спектра технологій робить C++ Builder універсальним інструментом створення програмних проєктів якого завгодно рівня складності. Навіть початкові користувачі зможуть швидко створювати програмні проєкти в C++ Builder, які матимуть професійний віконний інтерфейс найрізноманітнішої спрямованості, від суто обчислювальних та логічних, до графічних та мультимедійних. Інструментальні засоби інтерактивного середовища дозволяють студентам педагогічних вузів створювати і навчальні програми по вивченню різних предметів, зокрема інформатики, що згодом можуть бути використані ними, як майбутніми вчителями, у школі. Запропонований у статті матеріал дає уяву студентам, як приступати до створення таких програмних продуктів.

### СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Страуструп, Б. (2015). *Язык программирования C++*. Москва: БИНОМ.
2. Вайсфельд, М. (2005). *Объектно-ориентированный подход: Java, .Net, C++*. Москва: КУДИЦА-ОБРАЗ.
3. Шилдт, Г. (2010). *C++: базовый курс*. Москва: Вильямс.
4. Холингворт, Дж., Сворт, Б., Кэшмэн, М. & Густавсон, П. (2004). *Borland C++ Builder 6. Руководство разработчика*. Москва: Издательство Вильямс.
5. Дудзяний, І. М. (2013). *Програмування мовою C++*. Частина 1: *Парадигма процедурного програмування*. Львів : ЛНУ імені Івана Франка.
6. Лазурчак, Л. В. & Вдовичин, Т. Я. (2017). *Інформатика. Програмування мовою C++: методичні вказівки до виконання лабораторних робіт*. Дрогобич: Видавничий відділ Дрогобицького державного педагогічного університету імені Івана Франка.
7. Гилберт, С. & Маккарти, Б. (2003). *Самоучитель Visual C++ в примерах*. Киев: ТИД ДС.
8. Наконечний, С. І. & Савіна, С. С. (2003). *Математичне програмування*. Київ: КНЕУ.
9. Загребельний, С. Л., Костіков, А. А. & Мірінській, В. Е. (2012). *Основи візуального програмування в середовищі Visual Studio 2010*. Краматорск, Україна: ДГМА.
10. Павловская, Т. А. (2008). *C/C++ Программирование на языке высокого уровня*. Санкт-Петербург: Издательство «Питер».
11. Павловська, Т. А. & Щупак, Ю. А. (2007). *C/C++ Структурне програмування. Практикум*. Санкт-Петербург: Видавництво «Питер».
12. Сопронюк, Т. М. (2009). *Технології візуального й узагальненого програмування в C++ Builder*. Чернівці: ЧНУ.
13. Грицюк, Ю. І. & Рак, Т. Є. (2011). *Об'єктно-орієнтоване програмування мовою C++*. Львів: Вид-во Львівського ДУ БЖД.
14. Глинський, Я. М., Анохін, В. Є. & Ряжська, В. А. (2011). *C і C++ Builder*. Львів: СПД Глинський.

### REFERENCES (TRASLATED AND TRANSLITERATED)

1. Strastrup, B. (2015). *C++ programming language*. Moscow: BINOM.
2. Weisfeld, M. (2005). *Object Oriented Approach: Java, .Net, C++*. Moscow: COOKING-IMAGE.
3. Shieldt, G. (2010). *C++: basic course*. Moscow: "Wilyams".

4. Holingworth, J., Sworth, B., Cashman, M. & Gustavson, P. (2004). *Borland C ++ Builder 6. Developer Guide*. Moscow: "Wilyams".
5. Dudziany, IM (2013). *Programming in C ++. Part 1: Paradigm of procedural programming*. Lviv: The Ivan Franko National University of Lviv.
6. Lazurchak, L. V. & Vdovychyn, T. Ya. (2017). *Computer Science. Programming in C ++: methodical instructions for laboratory work*. Drohobych: Publishing Department of the Ivan Franko Drohobych State Pedagogical University.
7. Gilbert, S. & McCarthy, B. (2003). *Tutorial Visual C ++ in the examples*. Kiev: TID DS.
8. Nakonechnyj, S. I. & Savina, S. S. (2003). *Mathematical programming*. Kiev: KNEU.
9. Zagrebelnyj, S. L., Kostikov, A. A. & Mirinskaya, V. E. (2012). *Fundamentals of Visual Programming in Visual Studio 2010*. Kramatorsk, Ukraine: DHSA.
10. Pavlovskaya, T. A. (2008). *C / C ++ Programming in a high level language*. St. Petersburg: Peter Publishing.
11. Pavlovskaya, T. A. & Shchupak, Yu. A. (2007). *C / C ++ Structural programming. Practice*. St. Petersburg: "Peter".
12. Sopronyuk, T. M. (2009). *Technologies of visual and generalized programming in C ++ Builder*. Chernivtsi: ChNU.
13. Gritsuk, Yu. I. & Rak, T. Ye. (2011). *Object-oriented programming in C ++*. Lviv: Publishing of Lviv DU BZHD.
14. Glinsky, Ya. M., Anokhin, V. Ye. & Ryazhskaya, V.A. (2011). *C and C++ Builder*. Lviv: SPD Glinsky.

Стаття надійшла до редакції 24.12.2018.

The article was received 24 December 2018.

**Tatiana Vdovychyn, Lyubov Lazurchak**

**The Ivan Franko Drohobych State Pedagogical University, Drohobych, Ukraine**

## **FEATURES OF STUDYING PROGRAMS BY FUTURE TEACHERS OF INFORMATICS**

The process of training future teachers of informatics involves the formation of knowledge and skills, as well as professional skills in the context of obtaining a future qualification of the "teacher of informatics". The study of programming provides basic fundamental concepts and methods for their further application in different disciplines, supports interdisciplinary connections throughout the student learning process and in further professional activities. This article is devoted to highlighting the features of programming the C ++ Builder medium for future IT teachers.

C ++ Builder is one of the most powerful systems that allows you to create, at the most advanced level, both individual Windows applications, as well as ramified systems designed to work on corporate networks and the Internet. This is a tool for the rapid development of applications on C ++ for Windows, which supports component-based programming. The article explores the possibilities of this system for the development of complex and effective applications for future teachers of informatics. The focus is on the visual design of the software interface, which simplifies the interaction of the software with the user and adds a user-friendly interface to a professional level. The result of the visual design is the skeleton of the future program, which already has the corresponding codes. The visual development tools provide the performance of reusable visual components, combined with advanced database access tools.

In the article the visual possibilities of the C ++ programming language for the future teachers of informatics are explored on the example of creating a calculator and a text editor. Simplifying the development of the C ++ Builder environment and supporting a wide range of technologies makes it

a versatile tool for creating software projects of a diverse orientation. The C ++ Builder environment allows future IT teachers to create a variety of training programs that can be used in their professional activities, including at school.

**Keywords:** future teachers of computer science, programming language, C ++ Builder, object-oriented programming, visualization, visual software.