

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

МЕТОДИЧНІ ВКАЗІВКИ
до лабораторної роботи
«Використання шрифтів у графічних програмах на C#»

з курсу «Математичні основи комп'ютерної графіки»
для студентів спеціальності 122 Комп'ютерні науки,
спеціалізація 122-04 Інформаційно-аналітичні системи та технології,
186 Видавництво та поліграфія, спеціалізація 186-01 Інформаційні
технології в медіаіндустрії

Затверджено редакційно-видавничою
радою університету,
протокол № 2 від 23.06.2016 року

Харків
НТУ «ХПІ»
2017

Методичні вказівки до лабораторної роботи «Використання шрифтів у графічних програмах на С#» з курсу «Математичні основи комп'ютерної графіки» для студентів спеціальності 122 Комп'ютерні науки, спеціалізація 122-04 Інформаційно-аналітичні системи та технології, 186 Видавництво та поліграфія, спеціалізація 186-01 Інформаційні технології в медіаіндустрії / уклад. : Л. Б. Кащеєв, С. В. Коваленко. – Харків : НТУ «ХПІ», 2017. – 16 с.

Укладачі: Л. Б. Кащеєв,
С. В. Коваленко

Рецензент Л. М. Любчик

Кафедра системного аналізу та інформаційно-аналітичних технологій

ВСТУП

Метою цієї лабораторної роботи є знайомство з методами використання зображень, що зберігаються в файлах шрифтів. Цей вид графічної інформації успішно використовується в деяких графічних редакторах (наприклад, в CorelDraw, Xara), однак, застосування шрифтів у програмних середовищах, зокрема – в С#, як правило, випадає з поля зору підручників і посібників з візуального програмування. Особливий інтерес являє собою підключення шрифтів без їх реєстрації в операційному середовищі.

1. ТЕОРЕТИЧНІ ОСНОВИ

Однією з основних причин успішного поширення операційного середовища Windows серед офісних споживачів у перші роки впровадження цієї системи була сукупність шрифтів. Нагадаємо, що в DOS-середовищах набраний текст сприймався як алфавітно-цифрова інформація. Через це форма виведення документа на друкування, а саме кількість рядків на сторінці і кількість символів у рядку, багато в чому залежала від принтера.

Поворотним моментом у цьому напрямку стало поширення принципу WYSIWYG (What You See Is What You Get). Основною ідеєю цього спрямування прикладних програм стало зображення на екрані в реальному розмірі і реальній конфігурації тексту та образотворчих елементів. При цьому зі 100-відсотковою впевненістю можна очікувати, що саме так підготовлений текст буде виведений принтером на папір. І хоча в першу чергу системи WYSIWYG з'явилися на комп'ютерах Apple Lisa (1983) і Macintosh (1984), велика частина користувачів познайомилася з нею в середовищі Windows.

Накреслення літер і графічних елементів в ОС Windows визначається файлами з папки C:\Windows\Fonts. Перелік найбільш поширених розширень файлів, які можна зустріти в цій папці, наведені у табл. 1.1.

Таблиця 1.1 – Файли шрифтів: розширення і опис

Розширення	Опис
.abf	Двійковий файл шрифту Adobe
.afm	Файл метричних даних шрифту Adobe
.bdf	Шрифт ASCII
.chr	Налаштування символів шрифтів Borland
.compositefont	Композитний файл шрифту Windows
.eot	Файл шрифту OpenType
.fnt	Файл шрифту Windows
.fon	Файл шрифту
.fot	Файл ресурсів шрифту
.otf	Шрифт OpenType
.pfb	Двійковий файл шрифту для друкування
.pfm	Файл метричних даних шрифту
.pfr	Файл ресурсів портативного шрифту
.ttc	Колекція шрифтів TrueType
.tte	Файл даних шрифту Private Character Editor
.ttf	Шрифт TrueType
.vfb	Файл шрифту FontLab Studio
.xfn	Файл шрифту принтера Corel Ventura

Спочатку (до установки на машину додатків фірм Adobe, Borland, Corel, ChiWriter та ігор) у папці Fonts зберігаються в основному файли fnt, ttf і otf.

На цей час система Windows перейшла від однобайтового кодування символів (ANSI) до двобайтового (Unicode). В результаті нові шрифтові файли (в основному otf) містять зображення літер відразу в конфігурації латинських, кирилических, західноєвропейських, арабських, єврейських букв тощо. Це зробило недоцільним використання цілого ряду файлів шрифтів. Проте і раніше існувало багато специфічних конфігурацій літер, які не входили до стандартних шрифтових наборів (рис. 1.1 і рис. 1.2).



Рисунок 1.1 – Один з варіантів класифікації видів шрифтів, що взятий з сайту-каталогу шрифтів для вільного завантаження

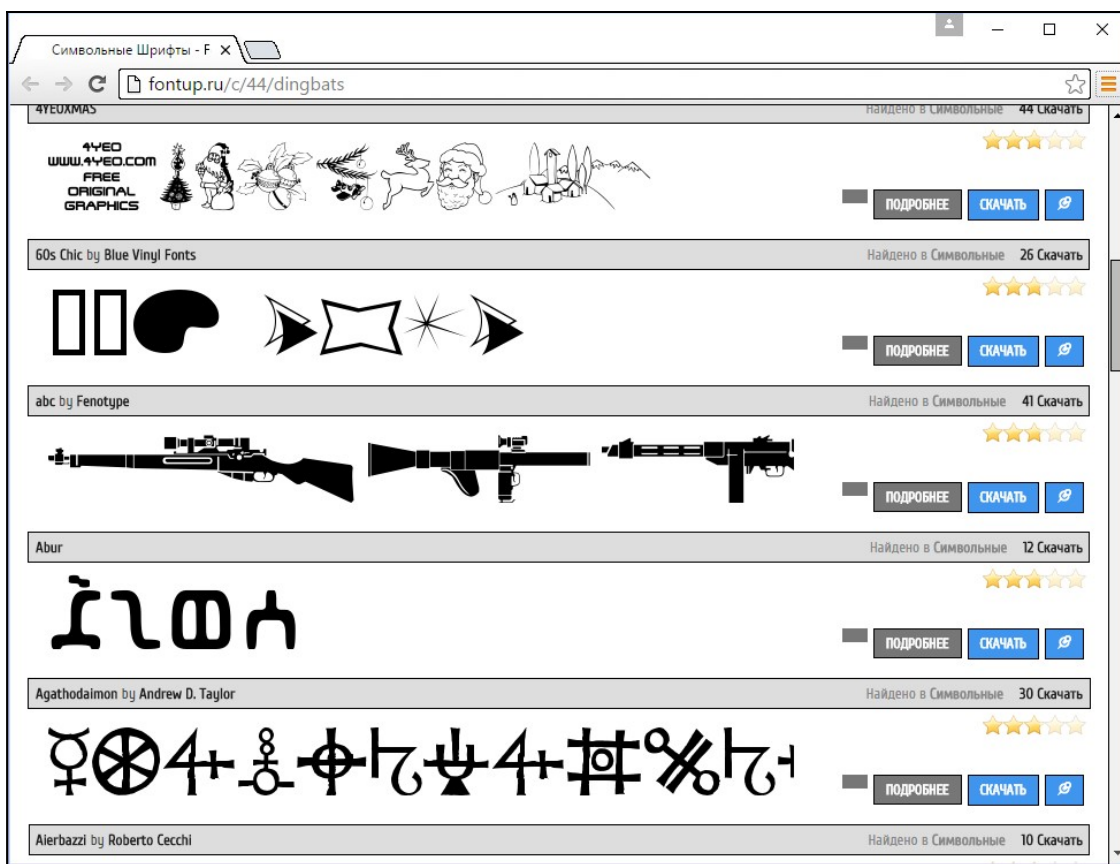


Рисунок 1.2 – Сторінка сайту вільного завантаження шрифтів. Спеціально обрана сторінка з символічними фонтами, щоб показати їх відмінність від шрифтів, які встановлюються у Windows за замовчуванням

2. УСТАНОВКА ШРИФТІВ

Установка шрифту на машину на сьогодні гранично полегшена.

Файл шрифту копіюється на диск і з адміністративними правами переноситься до папки C:\Windows\Fonts. Однак подібний підхід має два неприємні моменти. По-перше, при втраті інтересу до шрифту або деінсталяції відповідної програми, яка його встановила, файл у папці залишається. По-друге, при завантаженні операційного середовища для всіх файлів у папці Fonts створюється тимчасовий ресурсний файл (fot). Це веде до тривалого завантаження і займає пам'ять.

У ряді випадків при незначних змінах файл шрифту перейменовується, що вносить плутанину в питання, який шрифт інсталюваний, а який – ні. Тому перерахуємо стандартні шрифти, які при інсталяції встановлюються в усіх операційних середовищах (табл. 2.1).

Таблиця 2.1 – Шрифти, загальні для різних версій Windows і Mac

Назва шрифту в Windows	Назва шрифту в Mac	Сімейство
Arial	Arial, Helvetica	<i>sans-serif</i>
Arial Black	Arial Black, Gadget	<i>sans-serif</i>
Comic Sans MS	Comic Sans MS	<i>cursive</i>
Courier New	Courier New, Courier	<i>monospace</i>
Georgia	Georgia	<i>serif</i>
Impact	Impact, Charcoal	<i>sans-serif</i>
Lucida Console	Monaco	<i>monospace</i>
Lucida Sans Unicode	Lucida Grande	<i>sans-serif</i>
Palatino Linotype, Book Antiqua	Palatino	<i>serif</i>
Tahoma	Geneva	<i>sans-serif</i>
Times New Roman	Times	<i>serif</i>
Trebuchet MS	Helvetica	<i>sans-serif</i>
Verdana	Verdana, Geneva	<i>sans-serif</i>
Symbol	Symbol	-
Webdings	Webdings	-
Wingdings	Zapf Dingbats	-
MS Sans Serif	Geneva	<i>sans-serif</i>
MS Serif	New York	<i>serif</i>

У табл. 2.1 перераховані шрифти, загальні для всіх актуальних на цей момент операційних систем Windows, та їх еквіваленти в Mac OS. Такі шрифти іноді називають «безпечними шрифтами для браузерів» (browser safe fonts). Важливим моментом є і однакове їх відображення в інтерфейсі кросплатформених додатків.

3. ПОСТАНОВКА ЗАДАЧІ

У цій лабораторній роботі пропонується створити програмне застосування, яке буде виводити поточний час у різних містах світу на фоні вибраних зображень наручних годинників.

Виведення часу (годин, хвилин, секунд) здійснюється з урахуванням тимчасового зсуву і з підбором кольорового фону відповідно до кольору циферблата.

Виведення часу здійснюється шрифтом, що імітує рідкокристалічний 7-сегментний індикатор. Числове значення годин може перебувати в межах від 00 до 23. Числові значення виводяться двома знаками (якщо число менше 10, спереду дописується цифра 0).

4. АЛГОРИТМ ПІДГОТОВКИ РИСУНКА ДЛЯ ФОНУ

Як фон для виведення часу використані фотографії реальних наручних годинників. З цієї причини до виведення на форму зону виведення на рисунку пропонується «зачистити». При цьому слід розуміти, що на реальній фотографії фон-циферблат може мати трохи неоднотонний фон. Тому замість стирання або заливки пропонується виділити вільний фрагмент зображення і розтягнути його на всю частину циферблата, яку передбачається очистити (рис. 4.1).



Рисунок 4.1 – Очищення рисунка для фону: *а* – завантаження зображення; *б* – виділення порожнього фрагмента; *в* – розтяжка за розміром

5. ПРОГРАМНЕ ЗАВАНТАЖЕННЯ ШРИФТУ

Для програмного завантаження шрифту створимо застосування в середовищі C#. Помістимо файли DS-DIGI.TTF, DS-DIGIB.TTF, DS-DIGIU.TTF і DS-DIGIT.TTF у ту ж папку, де знаходиться *exe*-файл (наприклад, *WindowsFormApplication2.exe*). Помістимо на порожню форму візуальні компоненти *Button* і *Label*. Впишемо в опис форми і в метод «кнопки» кілька рядків коду, щоб перевірити, як зареєструється наш шрифт.

```
namespace WindowsFormsApplication2
{
    public partial class Form1 : Form
    {
        public System.Drawing.Text.PrivateFontCollection f;
        public Form1()
        {
            InitializeComponent();
            f=new System.Drawing.Text.PrivateFontCollection();
            f.AddFontFile("ds-digi.ttf");
        }
        private void button1_Click(object sender, EventArgs e)
        {
            Label1.Font=new Font(f.Families[0],24);
            Label1.Text="0 1 2 3 4 5 6 7 8 9";
        } } }
```

Після запуску форми на виконання натисканням кнопки ми повинні отримати форму такого вигляду, як на рис. 5.1.

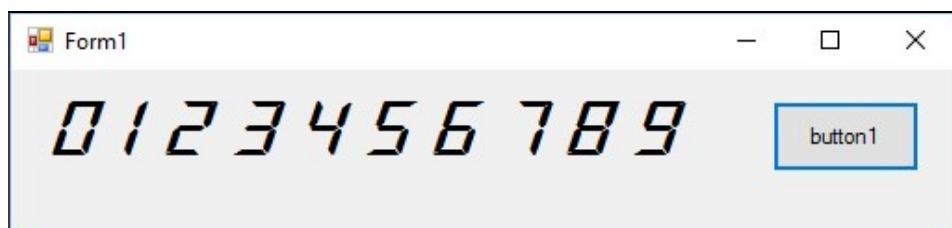


Рисунок 5.1 – Виведення цифр шрифтом DS-DIGI

Перевіряти виведення букв цим шрифтом не має сенсу: 7-сегментне оформлення літер в ньому виконано тільки для цифр.

6. ВИБІР РИСУНКА ДЛЯ ФОНУ

У кожному варіанті завдання на лабораторну роботу видається три різні фотографії наручних годинників. Кожна фотографія подається в двох файлах – розміром 200×248 і 400×496 пікселів. Мала фотографія використовується для виведення як `image` («наклейки») на кнопки. Велика фотографія (яка була відредагована в попередньому параграфі) призначена для виведення в `PictureBox` на формі.

Таким чином, на форму виводиться три кнопки з відповідними рисунками, в функцію-метод кожної кнопки вписується оператор завантаження відповідної картинки в `PictureBox`:

```
private void button1_Click(object sender, EventArgs e)
{
    pictureBox1.Load("watch01.png");
}
```

Насправді все буде трохи складніше. Усі фонові годинники трохи відрізняються за розміром, тому при перемиканні циферблата доведеться змінити місце розташування компонентів `label1` і `label2`, розмір символів, фоновий колір компонентів `label1`, стиль тексту – «жирний», «похилий» тощо. Тому вміст кнопок буде приблизно таким:

```
private void button1_Click(object sender, EventArgs e)
{
    pictureBox1.Load("watch01.png");
    label1.BackColor=Color.FromArgb(255, 205, 205, 205);
    label2.BackColor=Color.FromArgb(255, 197, 197, 197);
    label1.Left = 785;
    label1.Top = 247;
    label2.Left = 902;
    label2.Top = 260;
    label1.Font = new Font(f.Families[0], 34,
                          FontStyle.Bold);
    label2.Font = new Font(f.Families[0], 24,
                          FontStyle.Bold | FontStyle.Italic);
}
```

7. КОЛІР ФОНУ КОМПОНЕНТІВ LABEL

За замовчуванням компонент Label виводить літери чорним кольором на білому фоні. Для того, щоб напис не виділявся на фоновому рисунку необхідно налаштувати відповідний колір.

Робиться це так. Завантажити фоновий рисунок у будь-який графічний редактор, наприклад у Paint.

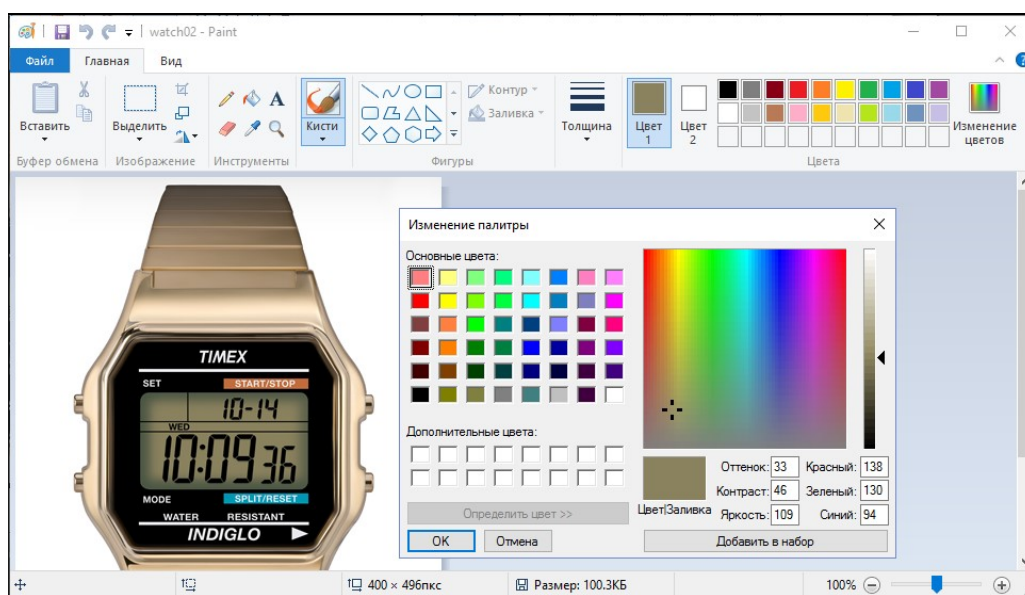


Рисунок 7.1 – Проба кольору інструментом «Палітра» та натискання кнопки «Зміна кольорів» у вікні палітри можна визначити колірні складові фонового кольору

За трьома складовими – Red, Green і Blue – можна налаштувати колір фону компонента, що досягається за допомогою такої команди:

```
label1.BackColor = Color.FromArgb(255, 138, 130, 94);
```

У поданому тут методі FromArgb() чотири аргументи: перший – прозорість (0 – прозорий, 255 – непрозорий), далі йдуть значення червоної, зеленої та синьої складової кольору.

8. ВИВЕДЕННЯ ЧАСУ ЗА ТАЙМЕРОМ

У роботі передбачається, що поточний час буде постійно оновлюватися при виведенні на форму. Це досягається за допомогою компонента Timer: програмне застосування з частотою таймера опитує системний годинник і виводить відповідні значення на форму:

```

private void timer1_Tick(object sender, EventArgs e)
{
    int h, m, s;
    h = DateTime.Now.Hour+sdvig;
    m = DateTime.Now.Minute;
    s = DateTime.Now.Second;
    string sh, sm, ss;
    sh=Convert.ToString(h); sm=Convert.ToString(m);
    ss = Convert.ToString(s);
    label1.Text = sh + ":" + sm;
    label2.Text= ss;
}

```

Після запуску форми на виконання натисканням кнопки отримаємо форму такого вигляду, як на рис. 8.1.

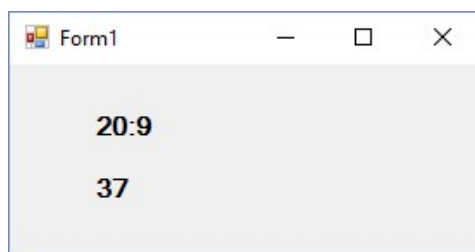


Рисунок 8.1 – Виведення за таймером поточного часу – годин, хвилин і секунд на форму

У наведеному фрагменті програми відсутня перевірка на значення годин більше 23 або менше 0, не дописується другий знак при виведенні, якщо значення складається з однієї цифри.

Важливе зауваження: винесений на форму таймер у середовищі C# неактивовано. Щоб його включити, треба надати значення True властивості Enabled.

У наведеному тексті програми змінна `sdvig` – це зсув часу з урахуванням часового поясу (він указаний у варіантах завдання на лабораторну роботу).

9. ВАРІАНТИ ЗАВДАНЬ НА ЛАБОРАТОРНУ РОБОТУ

Спільна частина завдання: виконати на формі приблизно такий годинник (рис. 9.1).

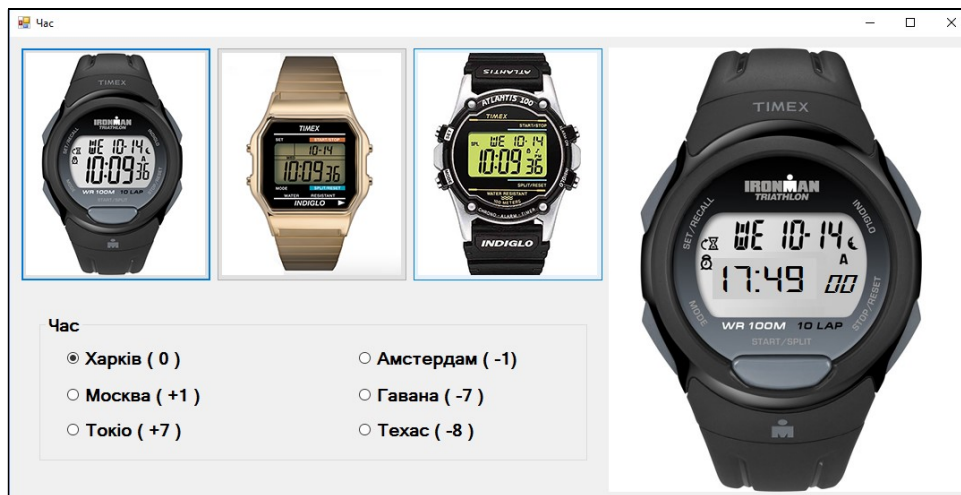


Рисунок 9.1 – Вигляд основної форми готової лабораторної роботи

Годинники опираються на системний час комп'ютера. Шрифт імпортується в програмне застосування.

При виконанні роботи передбачити, щоб на годиннику значення годин були не більше 23 (24 години – це 00). Години не можуть бути від'ємними (якщо за зрушенням часу виходить, наприклад, –3 години, відобразити 21). Години, хвилини, секунди з однієї цифри виводяться з лідируючими нулями (не 8:9, а 08:09).

Варіант 1

У радіокнопках розміщені міста для перемикання (у завданні наведені значення, на скільки їх час відрізняється від харківського): Харків 0, Москва +1, Токіо +7, Амстердам –1, Гавана –7, Остін (Техас) –8. Як альтернатива фону у цьому варіанті використовуються картинки watch01, watch02 і watch03.

Варіант 2

У радіокнопках розміщені міста для перемикання (у завданні наведені значення, на скільки їх час відрізняється від харківського): Київ 0, Москва +1, Анкоридж (Аляска) –11, Мадрид –1, Сан-Пауло –4, Рейк'явік –3. Як альтернатива фону у цьому варіанті використовуються картинки watch04, watch05 і watch06.

Варіант 3

У радіокнопках розміщені міста для перемикання (у завданні наведені значення, на скільки їх час відрізняється від харківського):

Харків 0, Ростов +1, Токіо +7, Абу-Дабі +2, Генуя –1, Мехіко –8. Як альтернатива фону у цьому варіанті використовуються картинки watch07, watch08 і watch09.

Варіант 4

У радіокнопках розміщені міста для перемикання (у завданні наведені значення, на скільки їх час відрізняється від харківського): Київ 0, Владивосток +8, Анкоридж (Аляска) –11, Барселона –1, Детройт –7, Монреаль –7. Як альтернатива фону у цьому варіанті використовуються картинки watch10, watch11 і watch12.

Варіант 5

У радіокнопках розміщені міста для перемикання (у завданні наведені значення, на скільки їх час відрізняється від харківського): Харків 0, Москва +1, Пекін +6, Амстердам –1, Сантьяго –5, Сеул +7. Як альтернатива фону у цьому варіанті використовуються картинки watch13, watch14 і watch15.

Варіант 6

У радіокнопках розміщені міста для перемикання (у завданні наведені значення, на скільки їх час відрізняється від харківського): Київ 0, Мінськ +1, Бангкок +5, Мадрид –1, Новосибірськ +4, Панама –7. Як альтернатива фону у цьому варіанті використовуються картинки watch16, watch17 і watch18.

Контрольні запитання

1. Які розширення бувають у файлів шрифтів?
2. В якій папці зберігаються шрифти, які встановлені в системі Windows?
3. Чим відрізняються растрові, векторні і постскріптовські шрифти?
4. Назвати колір, який складається з таких трьох колірних складових:
R=255, G=128, B=0 – ?
R=127, G=128, B=0 – ?
R=0, G=128, B=128 – ?
5. У C# застосована однорідна заливка (SolidBrush). Установка

кольору виконана за допомогою методу `Color.FromArgb()`. Чим відрізняються варіанти – `Color.FromArgb(255,0,0,128)` і `Color.FromArgb(128,0,0,128)`?

6. Як у редакторі Paint встановити колір заливки, який не відповідає жодному із стандартних кольорів палітри?

7. Як буде виводиться графічне зображення на кнопку (Button), якщо розмір рисунка значно перевершує розміри кнопки?

8. Візуальний компонент `PictureBox` має властивість `SizeMode`, яка може набувати значень `Normal`, `StretchImage`, `AutoSize`, `CenterImage`, `Zoom`. Пояснити, як зміниться вигляд зображення в `PictureBox` залежно від значення властивості `SizeMode`?

9. Як буде виводиться зображення в `PictureBox` у режимі `Normal`, якщо воно за розмірами менше розмірів `PictureBox`?

10. У `PictureBox` виводиться зображення, одне-єдине на весь сеанс роботи програми. Чим відрізняється завантаження файлу зображення на етапі проектування (властивість `Image` у компонента `PictureBox`) і завантаження в ході виконання програми – `pictureBox1.Load("a.bmp")`?

11. Які формати графічних файлів підходять для завантаження в компонент `PictureBox` без додаткових бібліотек?

12. Як у редакторі Paint дізнатися числові значення RGB – складових кольору точки растрового зображення?

СПИСОК ЛІТЕРАТУРИ

1. Кащеев Л. Б. Основи комп'ютерної графіки: навч. посіб. / Л. Б. Кащеев, С. В. Коваленко. – Харків : «Ранок», 2011. – 160 с.

2. Гросс К. С# 2008 и платформа .NET 3.5 FRAMEWORK: базовое руководство / К. Гросс. – М. : ООО «И. Д. Вильямс», 2009. – 480 с.

3. Петров М. Н. Компьютерная графика / М. Н. Петров, В. П. Молочков. 2-е изд.: – СПб. : Питер, 2006. – 811 с.

4. Рубанцев В. Программирование на языке С# 5.0. Компьютерная графика. Средний уровень / В. Рубанцев. – М. : RVGames, 2014. – 220 с.

Навчальне видання

МЕТОДИЧНІ ВКАЗІВКИ

до лабораторної роботи

«Використання шрифтів у графічних програмах на С#»

з курсу «Математичні основи комп'ютерної графіки»

для студентів спеціальності 122 Комп'ютерні науки,

спеціалізація 122-04 Інформаційно-аналітичні системи та
технології,

186 Видавництво та поліграфія, спеціалізація 186-01 Інформаційні
технології в медіаіндустрії

Укладачі:

КАЩЕСВ Леонід Борисович

КОВАЛЕНКО Сергій Володимирович

Відповідальний за випуск О. С. Куценко

Роботу до видання рекомендував М. І. Безменов

Редактор Н. В. Верстюк

План 2016 р., поз. 2

Підписано до друку 23.06.2016 р. Формат 60×84 1/16. Папір офсетний.

Друк – різнографія. Гарнітура Таймс. Ум. друк. арк. 3,7.

Наклад 50 прим. Зам. № 283-17. Ціна договірна

Видавничий центр НТУ «ХП».

Свідоцтво про державну реєстрацію ДК№ 3657 від 27.12.2009 р.

61002, Харків, вул. Фрунзе, 21

Друкарня «ФОП Пісня О. В.»

Свідоцтво про державну реєстрацію ВО2 № 248750 від 13.09.2007 р.

61002, Харків, вул. Гіршмана, 16а, кв. 21, тел. (057) 764-20-28