

ПРОГРАМА ПЕРЕТВОРЕННЯ РАСТРОВИХ КОНТУРНИХ ЗОБРАЖЕНЬ НА ОСНОВІ ФРАКТАЛЬНИХ СІТОК

© Назаркевич М., Клюйник І., 2017

Розроблено програмне забезпечення для перетворення растрових контурних зображень на фрактальні криві Гільберта. Програму можна використовувати для захисту та перетворення зображень у друкованих документах. Формат виведення зображень є векторним.

Ключові слова: перетворення зображень, захист друкованих документів, крива Гільберта.

The software for transforming raster images to contour curves fractal Hilbert. The program can be used to protect and convert images in printed documents. The output format is a vector image.

Key words: image conversion, protection of printed documents, Hilbert curve.

Вступ

Необхідність, актуальність і законність використання нових програмних засобів захисту інформації підтверджено у Законі України “Про інформацію” [1]. Але незважаючи на постійно зростаючий електронний документообіг, кількість паперових документів продовжує зростати. Захист традиційних матеріальних носіїв – паперових документів – донині забезпечується насамперед організаційними заходами і надійністю паролів [2]. Зараз же, коли сучасна офісна техніка дає змогу практично миттєво отримати копії документів настільки ідеальної якості, що часом сам власник оригіналу плутає оригінал і копію, потрібні нові засоби захисту паперових документів. Розроблена у цій публікації програма перетворення зображення у зображення, захищене фрактальними сітками, є не особливо новим способом захисту, але проте є нетрадиційним і специфічним для практики ділового обігу.

Як відомо, в боротьбі з правопорушеннями, зокрема в сфері обігу документів, існує два підходи. Один з них – запобігання злочину. Якщо потенційний злочинець побачить, що йому економічно не вигідно або не під силу створити фальшивку, то, природно, він цього не робитиме.

У нашій країні високий ступінь захисту грошових знаків і бланків паспортів вимагає дуже високих витрат для виготовлення якісних фальшивок. Такі перешкоди створюють серйозні труднощі для здійснення злочинів, а неможливість повторення елементів захисту допомагає легко виявити підробки.

Справжній документ у цьому випадку передбачає наявність спеціального захисного елемента, який неможливо виявити без спеціальних методик і відповідного програмного забезпечення. На цих способах захисту від фальсифікації і зупинимось детальніше. Розробимо програму для перетворення растрових контурних зображень на зображення, захищені фрактальною сіткою.

Функціональна схема програми перетворення на векторні зображення засобами фрактальної геометрії

Програмне забезпечення має перетворювати будь-яке растрове зображення на векторний спосіб представлення інформації із застосуванням фрактальної геометрії. Результат роботи

програми забезпечуватиме захист для друкованої та електронної інформації. Програма має вибирати одне зображення з банку зображень та перетворювати вибране зображення на єдину криву, яка не перетинається та створює силует контуру. Захист полягає у тому, що застосовуємо криву, яка видозмінює свої координати за фракталом Гільберта. Спочатку декомпонують задачу на елементарні операції та описують виконання всього процесу. На цьому етапі структуруємо розроблення програмного забезпечення.

Для розроблення програмного забезпечення можна використовувати Java, мову C та похідні від C, можна будувати систему захисту на продуктах фірми Adobe. Класична мова C не має розвинених інструментів для роботи з графікою. У похідній від неї мові C++ є розвинуті графічні інструменти, зокрема OpenGL, але у цій мові існує багато інших проблем, що ускладнюють розроблення прикладного програмного забезпечення, яке би давало змогу отримати документи високої поліграфічної якості. Програми, розроблені мовою C#, здебільшого можуть працювати в ОС, у якій їх було створено. Adobe – закрита система, що створює перепони під час проектування. Застосуємо мову Java, оскільки вона має розвинуті можливості роботи з графікою. Перевагою Java є те, що програмний продукт, який на ній створений, не залежатиме від архітектури комп'ютера. Для роботи програмного забезпечення достатньо буде встановити відповідну Java-машину [3].

Концепція роботи ПЗ

На рис. 1 зображено методику роботи програмного забезпечення для перетворення растрового зображення на зображення, захищене фрактальною сіткою.

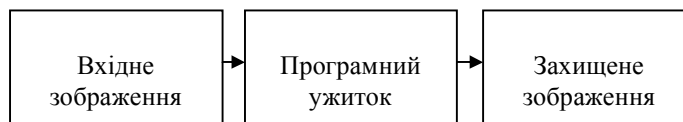


Рис. 1. Методика роботи ПЗ

На рис. 2 зображено структурну схему роботи програмного забезпечення.

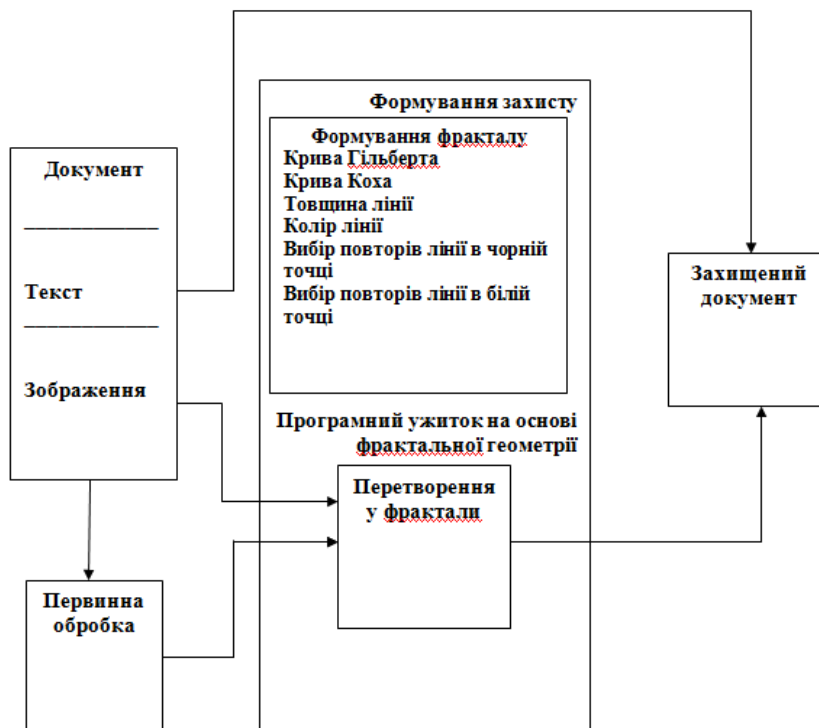


Рис. 2. Структурна схема роботи ПЗ

Для того, щоб перетворити растрове зображення на фрактальне, необхідно вибрати тип фракталу. Розрізняють три типи самоподібності у фракталах [4]: геометричні фрактали, алгебраїчні фрактали, стохастичні фрактали. Для побудови системи захисту доцільніше використати фрактали, які заповнюють усю площину документа. Тому алгебраїчні та стохастичні фрактали не можна використовувати під час розроблення цього ПЗ. Перетворення зображення на фрактальні криві розглядали в [5, 6]. Це дослідження є продовженням попередніх для розгляду побудови програмного забезпечення для створення захисту. Для розроблення класів програмного забезпечення перетворення растрових зображень на фрактальні криві використано ідеї, опубліковані у [7].

Криві Гільберта нагадують нитку, яку акуратно вкладено в одиничний квадрат. Алгоритм побудови кривих Гільберта є таким. Візьмемо квадрат зі стороною $\frac{1}{2}$, приберемо одну з його сторін і помістимо його точно в середину одиничного квадрата. На рис. 3 показано гільбертову криву 1-го порядку.

Зменшимо криву 1-го порядку рівно вдвічі і зробимо із неї 4 копії. Дві перемістимо, а дві інші теж перемістимо й повернемо на чверть оберту в протилежні боки. З'єднаємо кінці ліній трьома однаковими відрізками завдовжки як сторона нового зменшеного квадрата. На рис. 4 кожену копію забарвлено своїм кольором, а сполучені відрізки – чорні. Отримано криву 2-го порядку.

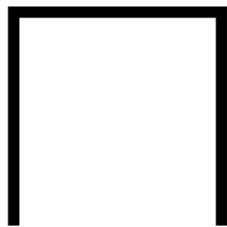


Рис. 3. Крива Гільберта 1-го порядку

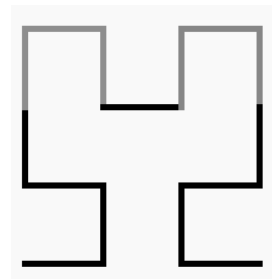


Рис.4. Крива Гільберта 2-го порядку

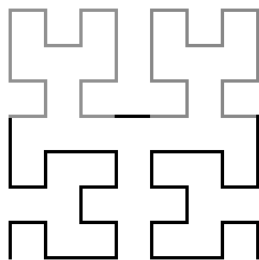


Рис. 5. Крива Гільберта 3-го порядку

Повторимо такі самі дії, але вже з отриманою ламаною лінією: зменшимо вдвічі, зробимо чотири копії, дві з яких повернемо, і знову з'єднаємо відрізками, які теж вкоротимо вдвічі. Повторювати цей алгоритм можна до нескінченності. На рис. 5 отримано криву Гільберта 3-го порядку.

Зазначимо, що крива Гільберта n порядку складається з $2^{2n} - 1$ відрізків завдовжки $1/2^n$.

Існують також і криві Пеано, в яких відсутні точки самоконтакту (так звані самоунікаючі криві). Одним із прикладів кривих такого роду є крива Госпера. Першим етапом побудови її є побудова ініціатора (рис. 4), який складається із семи однакових відрізків.

Захист на основі використання фрактальної геометрії при перетворенні графічних зображень

Для заповнення площини усього документа найпопулярнішими стали геометричні фрактали, які мають хаусдорфову розмірність 2 (криві Пеано) [4]. Також важливим фактором під час вибору саме цього типу фракталів є те, що документи та зображення є прямокутними, і для повноцінної їх обробки необхідний фрактал, що заповнює усю площину прямокутника. Відповідно інші фрактали, такі як криві Хартера–Хейтуея, крива Госпера не відповідають вимогам до остаточного вигляду друкованого документа. Для розроблення цього програмного забезпечення найдоцільніше використовувати криву Пеано, Серпінського, Гільберта або Мура.

Розроблення класів ПЗ для переведення растрової графіки у векторну

Програмне забезпечення містить візуальні компоненти, які показано на рис. 6. Їх розгорнутий вигляд у компонентах показано на рис. 7.

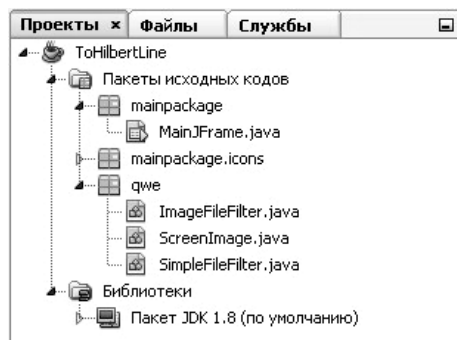


Рис. 6. Структура проекту

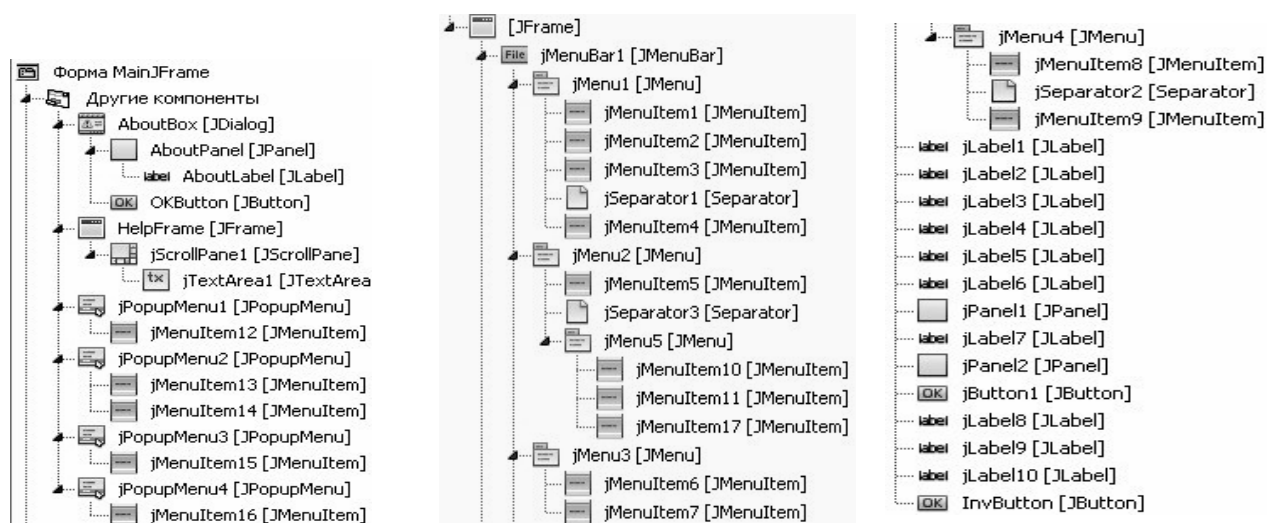


Рис. 7. Компоненти ПЗ

Опис класу MainJFrame

Клас MainJFrame є основним класом програми. У цьому класі реалізовано алгоритм побудови кривої Гільберта. На рис. 8 зображено алгоритм роботи програми drawHilbertCurve. На рис. 9 показано роботу методу blockIsWhite, який визначає поріг яскравості пікселя зображення. Реалізовано метод drawLine, який з'єднує лініями визначені раніше точки фракталу.

Опис класу ScreenImage

У класі ScreenImage знаходяться методи, що дають змогу зберігати у файл зображення. Компоненти класу ScreenImage зображено на рис. 10.

Основним завданням класу є збереження захищеного зображення і перетворення його на растр. Для такого програмного забезпечення розроблено спеціальний фільтр, завдяки якому у файловому діалозі показують лише потрібні формати файлів, а саме файли зображень з розширеннями .bmp, .gif, .GIF, .jpg, .jpeg, .JPG, .png, .PNG. Завдяки цьому фільтру користувач за замовчуванням матиме доступ лише до потрібних йому файлів і не зможе випадково обрати, наприклад, документ у форматі .txt або .psd, який програма не зможе коректно опрацювати.

Практична реалізація

Завантажуємо зображення (рис. 11).

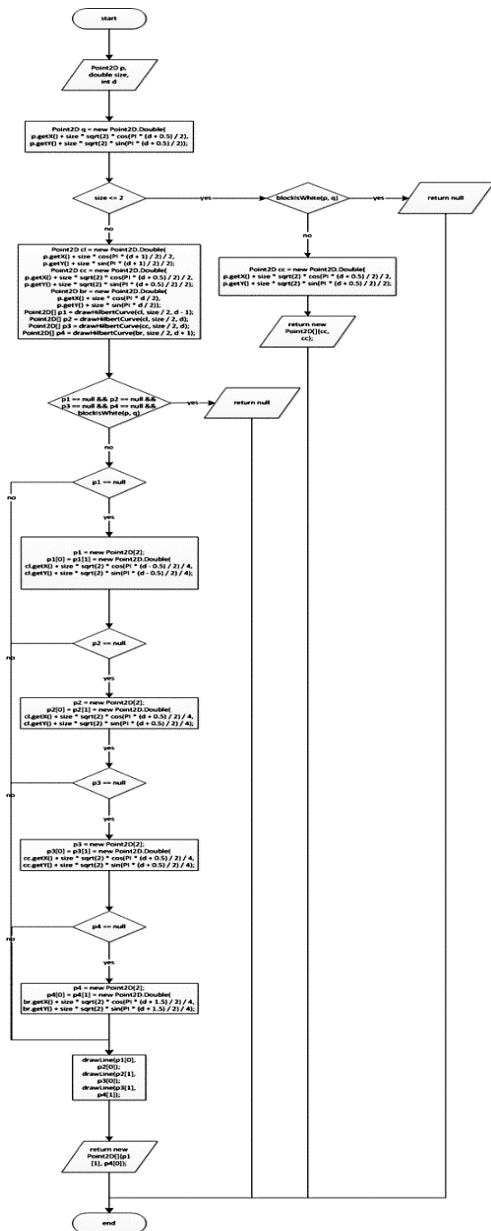
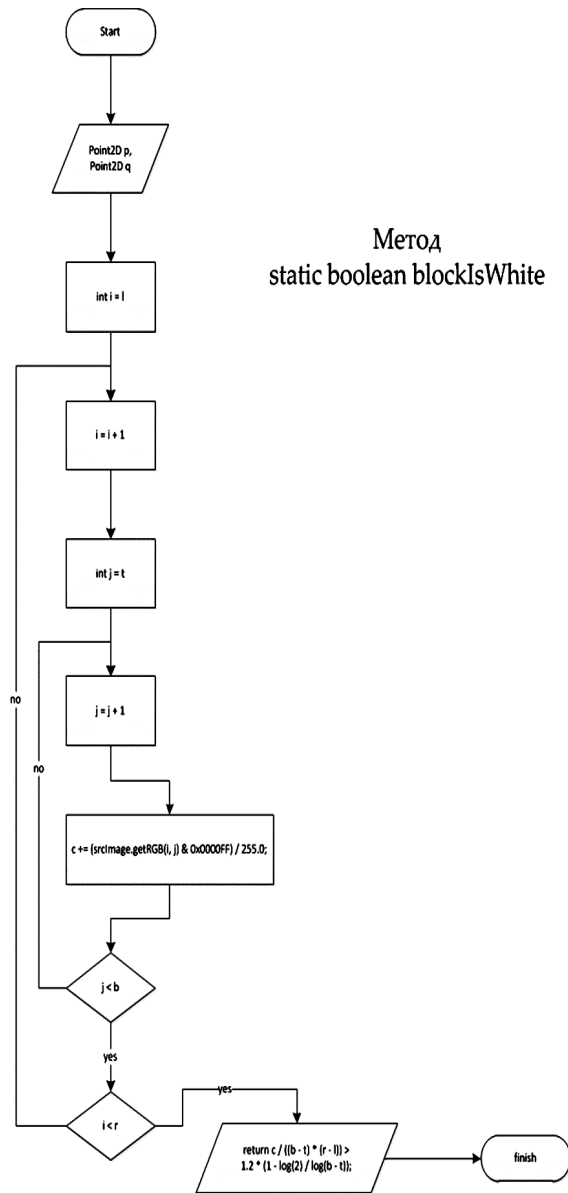


Рис. 8. Метод drawHilbertCurve



Метод
static boolean blockIsWhite

Рис. 9. Метод blockIsWhite

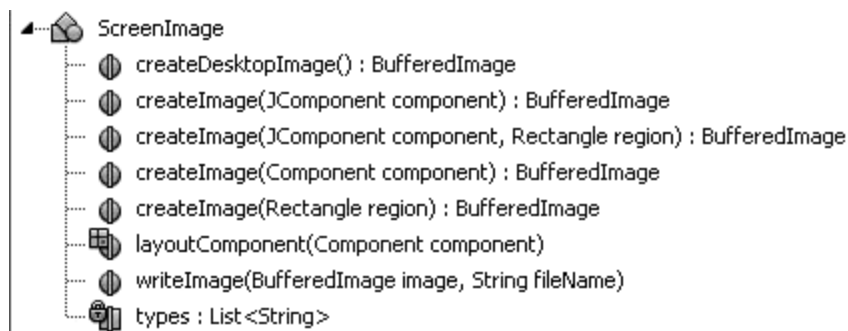


Рис. 10. Компоненти класу ScreenImage

Для коректної роботи програми зображення має бути квадратним. Програма може опрацювати зображення у будь-яких растрових форматах *.bmp, *.gif, *.jpg, *.png.



Рис. 11. Початкове зображення для перетворення

Результат роботи програми наведено на рис. 12. У програмі передбачено зміну кольору фону та кольору лінії кінцевого результату. За замовчуванням встановлено білий колір фону та чорний колір лінії. На рис. 13 показано результат роботи програми, який можна отримати при інверсії: чорному фоні та білій лінії.

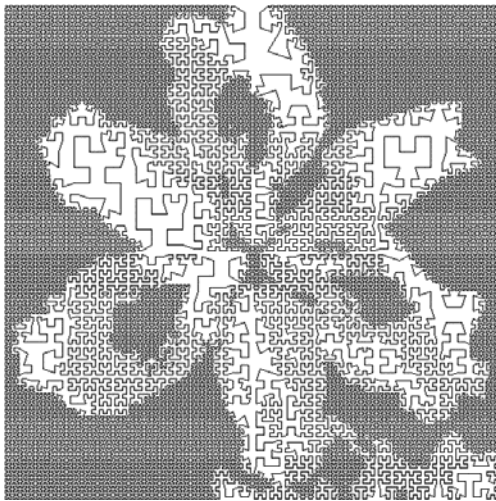


Рис. 12. Результат роботи програми.
Пряме зображення

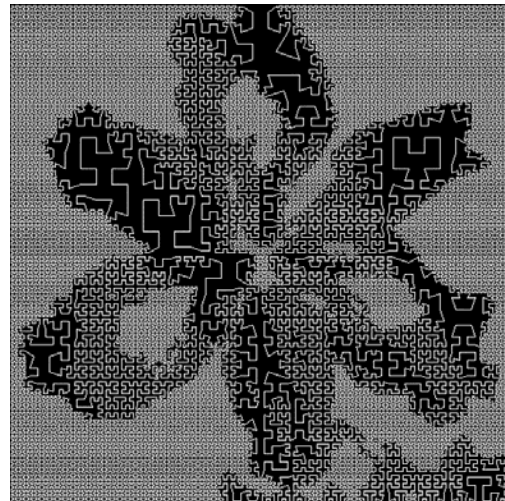


Рис. 13. Результат роботи програми.
Інверсія

Розроблення інтерфейсу

Одним з важливих елементів роботи програмного продукту є розроблення інтерфейсу. Проектуємо інтерфейс так, щоб він забезпечував максимальну зручність роботи користувача програми. Тому передбачаємо кнопки *Open*, *Save*, *Save As*, *Quit...* (завантаження), розміщені у меню *File*. Передбачено панель редагування (*Edit*), у якій є команди *Transform* (перетворити зображення) та *Set Color* (змінити колір). Зберегти остаточне зображення можна у форматах *.jpg*, *.png*, *.bmp*, *.gif*. У меню *View* можна переглянути ще раз початкове та перетворене зображення. Для зручності роботи розроблено допомогу для роботи з програмним продуктом, яку поміщено у меню за назвою *Help*.

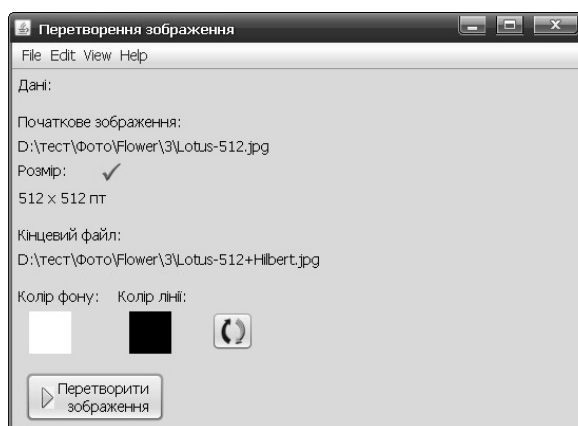


Рис. 14. Інтерфейс програми

Інтерфейс програми зображено на рис. 14.

Для того, щоб перетворити зображення за допомогою цієї програми, потрібно:

1. Вибравши пункт меню File -> Open, вибрати потрібне зображення (воно має бути квадратним).

2. Перетворити зображення, вибравши пункт меню Edit -> Transform або натиснувши кнопку “Перетворити зображення” на формі.

3. Якщо результат влаштовує, зберегти його, натиснувши File -> Save або Save As...

На формі виводяться дані про початкове та збережене остаточне зображення, а також про кольори. Їх можна змінювати, натиснувши правою кнопкою миші на потрібні дані. У програмі можна змінювати колір фону та лінії. Для зміни кольору фону виберіть меню Edit -> Set Color -> Background Color... Для зміни кольору лінії вибрати меню Edit -> Set Color -> Line Color... Для того, щоб поміняти кольори місцями, вибрати Edit -> Set Color -> Inverse. У меню View можна переглянути ще раз початкове та перетворене зображення.

Висновки

Проаналізовано задачі захисту друкованих документів, які може вирішити програмне забезпечення, побудоване на перетворенні графіки. Серед задач графіки виділено проблеми, які вирішують растровою, векторною та фрактальною графікою. Показано сфери застосування цих типів графіки. Виділено фрактальну графіку як найперспективний вид графіки для застосування у задачах захисту друкованих та електронних документів. Для розроблення ПЗ для перетворення на векторні зображення виділено засоби геометричних фракталів як такі, що дають можливість створити якісні засоби захисту.

1. Закон України “Про інформацію ” від 2 жовтня 1992 року (зі змінами) // Відомості Верховної Ради України. – 1992. – № 48. 2. Гостев И. М., Поволоцкий А. М. Защита традиционного документооборота нетрадиционными средствами // Информационно-методический журнал “Защита информации. Инсайд” № 3 май-июнь 2005 г. Режим доступа: http://www.inside-zh.ru/pages/3_2005/22.html. 3. Монахов В. В. Язык программирования Java и среда NetBeans. – 3-е изд., перераб. и доп. – СПб.: БХВ-Петербург, 2011. – 704 с.: ил. + DVD ISBN 978-5-9775-0671-7. 4. Морозов А. Д. Введение в теорию фракталов. – Москва-Ижевск: Институт компьютерных исследований, 2002. – 160 с. 5. Дронюк І. М., Рибалко Є. Метод захисту документів на основі фракталів // Вісник Держ. ун-ту “Львівська політехніка”. Комп’ютерні науки та інформаційні технології. – 2012. – № 732. – С.389 – 395. 6. Дронюк І. М., Квасниця С., Калінчук В. Формування захисних зображень на основі фрактальної геометрії // Вісник Держ. ун-ту “Львівська політехніка” Комп’ютерні науки та інформаційні технології. – № 751. – 2013. – С. 382–387. 7. Tsmots and O. Skorokhoda, “Methods and VLSI-structures for neural element implementation,” 2010 Proceedings of VIth International Conference on Perspective Technologies and Methods in MEMS Design, Lviv, 2010, pp. 135–135.