

УДК 002.5:004.45

ВИКОРИСТАННЯ JAVA - ТЕХНОЛОГІЙ ДЛЯ РОЗРОБКИ ГРАФІЧНИХ ДОДАТКІВ

ДРОБОТ В.В. ^{1*}, бакалавр,
ШИБКО О. М. ^{2*}, к.т.н, доц.

^{1*} Кафедра прикладної математики, Державний вищий навчальний заклад "Придніпровська державна академія будівництва та архітектури", вул. Чернишевського, 24-а, 49600, Дніпро, Україна, тел. +38 (0567) 56-34-10, e-mail: Vetalik_nexus@mail.ru

^{2*} Кафедра кафедра прикладної математики, Державний вищий навчальний заклад "Придніпровська державна академія будівництва та архітектури", вул. Чернишевського, 24-а, 49600, Дніпро, Україна, тел. +38 (0567) 56-34-10, email: prmat@mail.pgasa.dp.ua, ORCID ID: 0000-0001-5894-0642

Анотація. Мета. Метою роботи є розробка власного графічного редактора для малювання двовимірної графіки, розробка основних і базових інструментів для редагування графіки, використовуючи мову програмування Java і інтерфейси прикладного програмування Java 2D. А також використовуючи середу програмування NetBeans IDE 8.0.2 і IntelliJ IDEA 2016.1. **Методика** На сьогоднішній день графіка має важливе значення для мов програмування. Вона дозволяє візуалізувати будь-які програми, що надає яскравість і зручність для використання тих чи інших програм. Дозволяє створювати яскраві і цікаві Web-сторінки, зручна при розробці баз даних, написанні мобільних і комп'ютерних ігор. Так як ми живемо в тривимірному світі, то програмування двовимірної та тривимірної графіки на сьогоднішній день є актуальною і не можна залишити цю тему в стороні. **Результати.** Використовуючи класи Java 2D було створено простий графічний редактор JavaPaint, який дає можливість користувачеві створювати малюнки та редагувати графічні зображення. **Наукова новизна.** Представлено детальний аналіз усіх етапів розробки програмного забезпечення, розроблено покроковий план створення графічного додатку, розроблено власний дизайн програми за допомогою платформи NetBeans IDE., написано код програми. **Практична значимість.** Графіка має важливе значення для мов програмування. Вона дозволяє візуалізувати будь-які програми, що надає яскравість і зручність для використання тих чи інших програм. Дозволяє створювати яскраві і цікаві Web-сторінки, зручна при розробці баз даних, написанні мобільних і комп'ютерних ігор.

Ключові слова: графічне зображення, графічне програмування, Web-сторінка, Інтернет, Paint, Netbeans, IntelliJ IDEA, Java

ИСПОЛЬЗОВАНИЕ JAVA-ТЕХНОЛОГИЙ ДЛЯ РАЗРАБОТКИ ГРАФИЧЕСКИХ ПРИЛОЖЕНИЙ

ДРОБОТ В. В. ^{1*}, бакалавр,
ШИБКО О. Н. ^{2*}, к.т.н, доц.

^{1*} Кафедра прикладной математики, Государственное высшее учебное заведение "Приднепровская государственная академия строительства и архитектуры", ул. Чернышевского, 24-а, 49600, Днепр, Украина, тел. +38 (0567) 56-34-10, e-mail: Vetalik_nexus@mail.ru

^{2*} Кафедра прикладной математики Государственное высшее учебное заведение "Приднепровская государственная академия строительства и архитектуры", ул. Чернышевского, 24-а, 49600, Днепр, Украина, тел. +38 (0567) 56-34-10, email: prmat@mail.pgasa.dp.ua, ORCID ID: 0000-0001-5894-0642

Аннотация. Цель. Целью работы является разработка собственного графического редактора для рисования двухмерной графики, разработка основных и базовых инструментов для редактирования графики, используя язык программирования Java и интерфейсы прикладного программирования Java 2D. А также используя среду программирования NetBeans IDE 8.0.2 и IntelliJ IDEA 2016.1. **Методика.** На сегодняшний день графика имеет важное значение для языков программирования. Она позволяет визуализировать любые программы, предоставляет яркость и удобство для использования тех или иных программ. Позволяет создавать яркие и интересные Web-страницы, удобна при разработке баз данных, написании мобильных и компьютерных игр. Так как мы живем в трехмерном мире, то программирование двумерной и трехмерной графики на сегодняшний день является актуальной и нельзя оставить эту тему в стороне. **Результаты.** Используя классы Java 2D было создано простой графический редактор JavaPaint, который дает возможность пользователю создавать рисунки и редактировать графические изображения. **Научная новизна.** Представлен подробный анализ всех этапов разработки программного обеспечения, разработаны пошаговый план создания графического приложения, разработан собственный дизайн программы с помощью платформы NetBeans IDE., написан код программы. **Практическая значимость.** Представлен подробный анализ всех этапов разработки программного обеспечения, разработаны пошаговый план создания графического приложения, разработан собственный дизайн программы с помощью платформы NetBeans IDE., Написано код программы. Используя классы Java 2D создан простой графический редактор JavaPaint, который дает возможность пользователю создавать рисунки и редактировать графические изображения.

Ключевые слова: графическое изображение, графическое программирование, Web-страница, Интернет, Paint, Netbeans, IntelliJ IDEA, Java

USING JAVA-TECHNOLOGY TO DEVELOP GRAPHICAL APPLICATIONS

DROBOT V. V.^{1*}, *bachelor*,
SHIBKO O.N.^{2*}, *Cand. Sc. (Tech.), associate professor*

^{1*} Department of Heat Technique and Gas Supply, State Higher Education Establishment "Pridneprovsk State Academy of Civil Engineering and Architecture", 24-A, Chernishevskogo str., Dnipro 49600, Ukraine, phon. +38 (0567) 56-34-10, e-mail: Vetalik_nexus@mail.ru.

^{2*} Department of Heat Technique and Gas Supply, State Higher Education Establishment "Pridneprovsk State Academy of Civil Engineering and Architecture", 24-A, Chernishevskogo str., Dnipro 49600, Ukraine, phon. +38 (0567) 56-34-10, e-mail: prmat@mail.pgasa.dp.ua, ORCID ID: 0000-0001-5894-0642

Annotation. *The purpose.* The aim of the work is the development of a graphical editor for drawing two-dimensional graphics, the development of primary and basic tools for editing graphics using the Java programming language and Java 2D APIs. As well as using a programming environment NetBeans IDE 8.0.2 i IntelliJ IDEA 2016.1. *Methodology.* It is essential to the programming languages of today schedule. It allows you to visualize any program, provides brightness and ease of use for those or other programs. It allows you to create a bright and interesting Web-pages, useful when designing databases, writing mobile and computer games. Since we live in a three-dimensional world, the programming of the two-dimensional and three-dimensional graphics at present is relevant and can not leave this topic aside. *Results.* Using the Java 2D classes simple editor JavaPaint has been created, which enables the user to create drawings and edit graphics. *Scientific novelty.* A detailed analysis of all software development stages has been presented, a step by step plan for creating graphical applications has been developed, the own program design using NetBeans IDE platform has been developed, the program code has been written. *Practical significance.* . A detailed analysis of all software development stages has been presented, a step by step plan for creating graphical applications has been developed, the own program design using NetBeans IDE platform has been developed, the program code has been written. Using the Java 2D classes a simple editor JavaPaint has been created, which enables the user to create drawings and edit graphics.

Keywords: graphic images, graphical programming, Web-page, Internet, Paint, Netbeans, IntelliJ IDEA, Java

Введення

Останні кілька років розробники докладали масу зусиль, щоб інтегрувати графіку і анімацію в свої аплети і додатки Java. Однак спочатку включені в Java графічні пакети AWT Java мали обмежені засоби для вирішення таких завдань. Тепер же, використовуючи інтерфейси прикладного програмування Java 2D, розробники можуть реалізовувати набагато більш складні графічні додатки, включаючи ігри, зберігачі екрану, екранні заставки і тривимірний графічний інтерфейс.

Мета

Метою роботи є розробка власного графічного редактора для малювання двовимірної графіки, розробка основних і базових інструментів для редагування графіки, використовуючи мову програмування Java і інтерфейси прикладного програмування Java 2D. А також використовуючи середу програмування NetBeans IDE 8.0.2 i IntelliJ IDEA 2016.1.

Методика

На сьогоднішній день графіка має важливе значення для мов програмування. Вона дозволяє візуалізувати будь-які програми, що надає яскравість і зручність для використання тих чи інших програм. Дозволяє створювати яскраві і цікаві Web-сторінки, зручна при розробці баз даних, написанні мобільних і комп'ютерних ігор.

Так як ми живемо в тривимірному світі, то програмування двовимірної та тривимірної графіки на сьогоднішній день є актуальною і не можна залишити цю тему в стороні.

Почнемо з розгляду поетапної розробки дизайну програми та деяких графічних можливостей Java 2D і поступово перейдемо до суті самої розробки.

Розробка програми - це не лише написання програми. Написання програми [1] є одним з етапів. Етапами розробки програм є:

- Формулювання та аналіз фізичної задачі
- Складання математичної моделі
- Складання алгоритму задачі
- Створення програми
- Складання тексту програми
- Введення тексту програми в комп'ютер
- Синтаксична налагодження програми
- Тестування і семантична налагодження
- Документування програми
- Запуск готової програми та аналіз отриманих результатів.

Після досконалого теоретичного аналізу етапів розробки, можна переходити до практичної частини створення проекту.

Щоб показати графічні можливості мови Java, було прийнято рішення розробити графічний редактор зі стандартним набором інструментів.

Для створення і написання програми доведеться познайомитися з програмними комплексами в яких

ми будемо працювати, їх ще називають платформою для розробки. Так як в результаті потрібно отримати повноцінний продукт написаний на мові програмування Java, з різними інструментами та декількома кнопками, потрібно створювати форму на якій буде розташовуватися весь користувацький інтерфейс.

Інтерфейс користувача (англ. user interface, UI, дружній інтерфейс) — засіб зручної взаємодії користувача з програмою. Сукупність засобів для обробки та відображення інформації, максимально пристосованих для зручності користувача; у графічних системах інтерфейс користувача реалізується багато віконним режимом, змінами кольору, розміру, видимості (прозорість, напівпрозорість, невидимість) вікон, їхнім розташуванням, сортуванням елементів вікон, гнучкими налаштуваннями як самих вікон, так і окремих їхніх елементів (файли, папки, ярлики, шрифти тощо), доступністю багатьох налаштувань.

Проаналізувавши існуючі платформи [3], найбільш зручними для створення програми мовою Java виявилися дві: NetBeans IDE 8.0.2 і IntelliJ IDEA 2016.1.

Перша платформа NetBeans IDE 8.0.2 зручніша для створення форм, а друга IntelliJ IDEA 2016.1 зручніша для написання коду вручну і більш сприятлива при відладці програми.

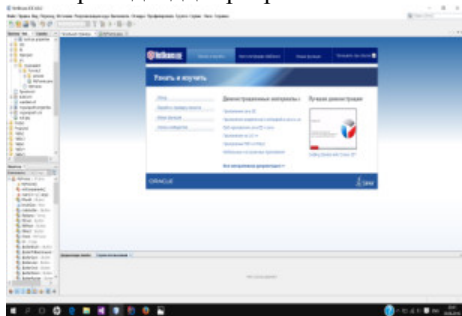


Рис. 1. Платформа NetBeans IDE 8.0.2 / Platform NetBeans IDE 8.0.2

Java Development Kit, скорочено JDK — безкоштовний, розповсюджуваний Oracle (раніше Sun) комплект розробника додатків на мові Java, який включає до себе компілятор Java (javac), стандартні бібліотеки класів Java, приклади, документацію, різноманітні утиліти і виконавчу систему Java (JRE). В склад JDK не входить інтегроване середовище розробки на Java (IDE), тому розробник, що використовує тільки JDK, повинен використовувати текстовий редактор і компілювати та виконувати свої програми через утиліти командного рядка.

Усі середовища розробки на Java, такі, як Eclipse, Netbeans, IntelliJ IDEA, Android Studio, спираються на сервіси JDK, що надаються, і викликають для компіляції Java-програм компілятор з комплекту JDK (javac). Тому ці середовища розробки або включають в комплект постачання одну з версій JDK або вимагають для своєї роботи попереднє встановлення JDK на машині розробника.

Повні вихідні тексти JDK, включаючи вихідні тексти самого Java-компілятора (javac) вільнодоступні.

SDK (від англ. Software development kit) - комплект засобів розробки, який дозволяє фахівцям з програмного забезпечення створювати додатки для певного пакету програм, програмного забезпечення базових засобів розробки, апаратної платформи, комп'ютерної системи, ігрових консолей, операційних систем і інших платформ.

Програми, тісно взаємодіючи з користувачем, що сприймають сигнали від клавіатури і миші, працюють в графічному середовищі. Кожна програма, призначена для роботи в графічному середовищі, повинна створити хоча б одне вікно, в якому буде відбуватися його робота, і зареєструвати його в графічній оболонці операційної системи, щоб вікно могло взаємодіяти з операційною системою і іншими вікнами: перекриватися, переміщатися, змінювати розміри, згортатися в ярлик.

В технології Java графіка ускладнюється тим, що додатки Java повинні працювати в будь-якій або хоча б у багатьох графічних середовищах. Потрібна бібліотека класів, незалежна від конкретної графічної системи. У першій версії JDK завдання вирішили в такий спосіб: були розроблені інтерфейси, що містять методи роботи з графічними об'єктами. Класи бібліотеки AWT (Abstract Window Toolkit) реалізують ці інтерфейси для створення додатків. Програми Java використовують ці методи для розміщення і переміщення графічних об'єктів, зміни їх розмірів, взаємодії об'єктів.

Бібліотека класів Java, заснованих на реє(парних)-інтерфейсах, отримала назву AWT. При виведенні об'єкта, створеного в додатку Java і заснованого на реє-інтерфейсі, на екран створюється парний йому (peer-to-peer) об'єкт графічної підсистеми операційної системи, який і відображається на екрані.

У версії JDK 1.1 бібліотека AWT була перероблена. У неї додана можливість створення компонентів, повністю написаних на Java і не залежать від реє-інтерфейсів. Такі компоненти стали називати "легкими" (lightweight) на відміну від компонентів, реалізованих через реє-інтерфейси, названих "важкими" (heavy).

Була створена велика бібліотека "легких" компонентів Java, названа Swing. У ній були переписані всі компоненти бібліотеки AWT, так що бібліотека Swing може використовуватися самостійно, не дивлячись на те, що всі класи з неї розширюють класи бібліотеки AWT.

Бібліотека класів Swing поставлялася як доповнення до JDK 1.1. До складу Java 2 SDK вона включена як основна графічна бібліотека класів, що реалізує ідею "100% Pure Java", поряд з AWT.

У Java 2 бібліотека AWT значно розширена додаванням нових засобів малювання, виведення текстів і зображень, які отримали назву Java 2D, і

початкова версія додатку, тому в майбутньому є можливість доробити проект і зробити його багатомовним.

Користувачі графічного редактора різного віку, діти та дорослі, тому треба розробити інтерфейс додатку максимально простим, цікавим та інтуїтивним.

Кольори які потрібно використовувати для створення, повинні бути не різкими для ока, та м'якими тонами.

Готова фінальна форма виглядає так:

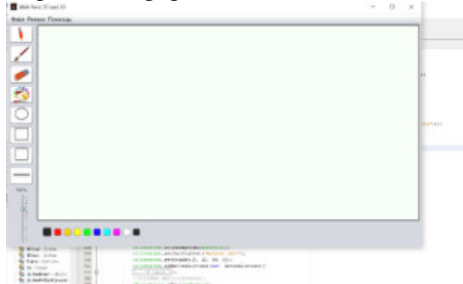


Рис. 4. Фінальне вікно додатку / The final application window

Коли користувач натискає пункт меню «Справка» відкривається файл документації по використанню графічного додатку:



Рис. 5. Меню «Справка» / Menu "Help"

При створенні компонента, тобто об'єкта класу Component, автоматично формується його графічний контекст (graphics context). У контексті розміщується область малювання і виведення тексту і зображень. Контекст містить поточний і альтернативний колір малювання і колір фону - об'єкти класу Color, поточний шрифт для виведення тексту - об'єкт класу Font.

У контексті визначена система координат, початок якої з координатами (0, 0) розташоване у верхньому лівому кутку області малювання, вісь Ox направлена вправо, вісь Oy - вниз. Точки координат знаходяться між пікселями.

Управляє контекстом клас Graphics або новий клас Graphics2D, введений в Java 2. Оскільки графічний контекст сильно залежить від конкретної графічної платформи, ці класи зроблені абстрактними. Тому не можна безпосередньо створити екземпляри класу Graphics або Graphics2D.

Однак кожна віртуальна машина Java реалізує методи цих класів, створює їх екземпляри для компонента і надає об'єкт класу Graphics методом

getGraphics () класу Component або як аргумент методів paint () і update ().

В системі пакетів і класів Java 2D, основа якої-клас Graphics2D пакета java.awt, є кілька принципово нових положень.

Крім координатної системи, прийнятої в класі Graphics і названої координатним простором користувача (User Space), введена ще система координат пристрою виведення (Device Space): екрану монітора, принтера. Методи класу Graphics2D автоматично переводять (transform) систему координат користувача в систему координат пристрою при виведенні графіки:

- перетворення координат користувача в координати пристрою можна задати "вручну", причому перетворенням здатне служити будь-яким перетворення площини, зокрема, поворот на будь-який кут і / або стиснення / розтягування. Воно визначається як об'єкт класу AffineTransform. Його можна встановити як перетворення за замовчуванням методом setTransform (). Можливо виконувати перетворення "на льоту" методами transform і translate і робити композицію перетворень методом concatenate ();

- оскільки таке перетворення матеріально, координати задаються речовими, а не цілими числами;

- графічні примітиви: прямокутник, овал, дуга та ін., реалізують тепер новий інтерфейс shape пакету java.awt. Для їх креслення можна використовувати новий єдиний для всіх фігур метод draw, аргументом якого здатний служити будь-який об'єкт, який реалізував інтерфейс shape. Введено метод fill, що заповнює фігури-об'єкти класу, який реалізував інтерфейс shape;

- для креслення (stroke) лінії введено поняття пера (pen). Властивості пера описує інтерфейс stroke. Клас BasicStroke реалізує цей інтерфейс. Перо володіє чотирма характеристиками:

1. Воно має товщину (width) в один (за замовчуванням) або кілька пікселів;

2. Воно може закінчити лінію (end cap) закругленням - статична константа CAP_ROUND, прямим обрізом - CAP_SQUARE (за замовчуванням), або не фіксувати певний спосіб закінчення - CAP_BUTT;

3. Воно може сполучати лінії (line joins) закругленням - статична константа JOIN_ROUND, відрізком прямої - JOIN_BEVEL, або просто стикувати - JOIN_MITER (за замовчуванням);

4. Воно може креслити лінію різними пунктиром (dash) і штрих-пунктиром, довжини штрихів і проміжків задаються в масиві, елементи масиву з парними індексами задають довжину штриха, з непарними індексами - довжину проміжку між штрихами.

- Методи заповнення фігур описані в інтерфейсі Paint. Три класи реалізують цей інтерфейс. Клас color реалізує його суцільний (solid) заливанням, клас GradientPaint - градієнтним (gradient) заповненням,

при якому колір плавно змінюється від однієї заданої точки до іншої заданої точки, клас `Texturepaint` - заповненням по попередньо заданому зразку (`pattern fill`);

- букви тексту розуміються як фігури, т. е. об'єкти, що реалізують інтерфейс `shape`, і можуть викреслюватися методом `draw` з використанням всіх можливостей цього методу. При їх кресленні застосовується перо, все методи заповнення і перетворення;

- крім імені, стилю і розміру, шрифт отримав багато додаткових атрибутів, наприклад, перетворення координат, підкреслення або перекреслення тексту, виведення тексту справа наліво. Колір тексту і його фону є тепер атрибутами самого тексту, а не графічного контексту. Можна задати різну ширину символів шрифту, надрядкові і підрядкові індекси. Атрибути встановлюються константами класу `TextAttribute`;

- процес візуалізації (`rendering`) регулюється правилами (`hints`), певними константами класу `RenderingHints`.

З такими можливостями Java 2D стала повноцінною системою малювання, виведення тексту і зображень.

Після довгого процесу налагодження коду, оптимізації програми, було виявлено багато помилок та прийнято до уваги, багато з яких виправлені, але деякі функції та недоліки занесені до списку подальшого покращення програмного продукту.

Для того щоб кінцева програма пройшла всі етапи тестування і отримані результати були достовірними, ми встановимо графічний редактор `JavaPaint` на персональні комп'ютери різних користувачів. В результаті тестування користувачі виявили недоліки, деякі з яких були виправлені.

Наприклад інструмент «Пензлик» з широкою товщиною малював квадратні лінії і всім шістьом

користувачам це здалося незручним, тому це було виправлено додавання методу `BasicStroke()`, що дало можливість створити пензлик з новими параметрами, який малює заокруглі лінії.

Результати

Використовуючи класи `Java2D` було створено простий графічний редактор `JavaPaint`, який дає можливість користувачеві створювати малюнки та редагувати графічні.

Наукова новизна та практична значимість

Представлено детальний аналіз усіх етапів розробки програмного забезпечення, розроблено покроковий план створення графічного додатку, розроблено власний дизайн програми за допомогою платформи `NetBeans IDE`, написано код програми. Використовуючи класи `Java2D` створено простий графічний редактор `JavaPaint`, який дає можливість користувачеві створювати малюнки та редагувати графічні зображення.

Висновки

Проаналізувавши, які функції необхідні нашому додатку, на прикладі усім відомого редактора «`MS Paint`»:

1. Розроблено власний дизайн програми за допомогою платформи `NetBeans IDE`.
2. Створено інтерфейс програми;
3. Показано графічні можливості `Java`
4. Написано код програми на мові програмування `Java`;
5. Створено простий графічний редактор `JavaPaint`, який дає можливість користувачеві створювати малюнки та редагувати графічні зображення `Java`.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Блох Джошуа. *Java. Эффективное программирование*/Дж. Блох. – Москва.: Лори, 2014. – 310 с.
2. Иванова Г.С. *Основы программирования*/ Г.С. Иванова.– Москва.:МГТУ им. Баумана, 2007. – 415 с.
3. Монахов В.В. *Язык программирования Java и среда NetBeans* /В.В. Монахов. – Санкт-Петербург.:БХВ-Петербург, 2012. – 704 с.
4. Свистунов А.Н. *Построение распределённых программных систем на Java* / А.Н. Свистунов. –Санкт- Петербург: Бином, 2014. – 280 с.
5. Файн Я. *Программирование на Java* / Я. Файн. – Москва.: Самиздат, 2011. - 231 с.
6. Эккель Б. *Философия Java* / Б.Эккель. – Санкт-Петербург: Питер, 2009. – 640 с.

REFERENCES

1. Bloch J. *Java. Effektivnoe programmirovaniye* [Effective programming]. Moscow : Lori, 2014. - 310 p. (in Russian).
2. Ivanova, G.S. *Osnovy programmirovaniya* [Introduction to Programming]. Moscow.: MSTU. Bauman, 2007. - 415 p. (in Russian).
3. Monakhov V.V *Yazyk programmirovaniya Java i sreda NetBeans* [Java programming language and the environment NetBeans]. St. Petersburg: BHV-Petersburg. 2012. - 704 p. (in Russian).
4. Svistunov A.N. *Postroenie raspredelennykh programmykh sistem na Java* [Construction of distributed software systems on Java]. St. Petersburg: Binom Publ., 2014, 280 p. (in Russian).
5. Fine Ya. *Programmirovaniye na Java* [Java programming]. Moscow: Samizdat Publ., 2011. - 231 p. (in Russian).
6. Eckel B. *Filosofiya Java* [Java philosophy]. St. Petersburg: Piter Publ., 2009. - 640 p. (in Russian).