

**ĐẠI HỌC QUỐC GIA TP.HỒ CHÍ MINH**

Trường đại học Công nghệ Thông tin

-----🙞🙜🕮🙞🙜-----

Logo

Description automatically generated

**BÁO CÁO MÔN HỌC**

**HỌC KỲ: 1 NĂM HỌC: 2022 – 2023**

**ĐỀ TÀI: PHẦN MỀM VẼ PAINT**

*Lớp:* CS511.N11

*Giảng viên hướng dẫn:*

ThS. Nguyễn Bích Vân

*Nhóm thực hiện:*

|  |  |  |
| --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** |
| 1 | Đỗ Minh Nhật | 21522415 |

*Thành phố Hồ Chí Minh, tháng 1 năm 2023*

**NHẬN XÉT CỦA GIẢNG VIÊN**

................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................................

**MỤC LỤC**

Nội dung

[**LỜI CẢM ƠN** 4](#_Toc124586850)

[**MỞ ĐẦU** 5](#_Toc124586851)

[**A.** **TỔNG QUAN NHÓM:** 6](#_Toc124586852)

[**I.** **Thông tin nhóm:** 6](#_Toc124586853)

[**II.** **Phương thức làm việc:** 6](#_Toc124586854)

[**B.** **BÁO CÁO ĐỒ ÁN:** 6](#_Toc124586855)

[**I.** **Giới thiệu chung: Đề tài đồ án “Chương trình Paint”** 6](#_Toc124586856)

[**II.** **Ý tưởng thực hiện:** 8](#_Toc124586857)

[**III.** **Thiết kế dữ liệu:** 11](#_Toc124586858)

[**IV.** **Thiết kế giao diện:** 34](#_Toc124586859)

[**TÀI LIỆU THAM KHẢO** 36](#_Toc124586860)

# **LỜI CẢM ƠN**

Lời đầu tiên em xin gửi lời cảm ơn đến ThS. Nguyễn Bích Vân, cô đã nhiệt tình giảng dạy trên lớp, hỗ trợ những thông tin cần thiết, giải đáp những thắc mắc và góp ý cho em cũng như các bạn khác trong suốt quá trình thực hiện đề tài. Đồng thời em cũng muốn cảm ơn các anh chị khóa trên, đặc biệt là anh chị trong khoa đã chia sẻ những kinh nghiệm quý báu về môn học cũng như những kiến thức liên quan. Cảm ơn bạn bè, anh chị đã tạo cho em điều kiện thuận lợi, đưa ra những nhận xét và góp ý chân thành, cũng như đã động viên, hỗ trợ em hoàn thành đề tài.

Do đây là lần đầu tiên em thực hiện một đồ án có thể gọi là lớn và quy mô, nhưng với lượng kiến thức và kỹ năng hạn chế nên không thể tránh khỏi những thiếu sót. Chính vì vậy em rất mong được nhận những góp ý nhằm hoàn thiện và trao dồi thêm nhiều kiến thức mới để làm hành trang thực hiện các đề tài khác trong tương lai.

Chân thành cảm ơn



# **MỞ ĐẦU**

Ngôn ngữ lập trình C# (C# Programming Language) là một môn học cung cấp cho sinh viên những kiến thức cơ bản về ngôn ngữ lập trình C#, hiểu được kiến trúc .NET, một kiến trúc rất được ưa chuộng và sử dụng phổ biến ngày nay. Qua môn học, ta có được kỹ năng lập trình trên môi trường Window như lập trình giao diện, lập trình đồ họa GDI+, xử lý tập tin và thư mục, kết nối cơ sở dữ liệu, lập trình đồng hành..., từ đó sinh viên có khả năng tự xây dựng 1 ứng dụng hoàn chỉnh ở mức độ vừa phải bằng ngôn ngữ C#. Môn học này cũng cung cấp cho sinh viên một số kiến thức cơ bản để có thể tự nghiên cứu các kỹ thuật lập trình sâu hơn trên môi trường Windows.

Trong khuôn khổ của môn học, chương trình Demo Paint Version 1.0 được xây dựng bởi sinh viên Đỗ Minh Nhật lớp Ngôn ngữ lập trình C# (CS511.N11) trường Đại học Công nghệ Thông tin ĐHQG HCM. Chương trình Demo Paint Version 1.0 xây dựng dựa vào các chức năng sẵn có của Microsoft Paint nhưng được cải tiến và có thêm một số chức năng nâng cao phục vụ tốt hơn cho yêu cầu ngày càng cao của người sử dụng.

Cùng với sự hướn dẫn của giảng viên ThS. Nguyễn Bích Vân, em đã hoàn thành được các yêu cầu từ cơ bản nhất đến nâng cao cho chương trình Demo Paint Version 1.0. Những kết quả thu hoạch được nhóm sẽ trình bày trong những phần sau của báo cáo này.

1. **TỔNG QUAN NHÓM:**
2. **Thông tin nhóm:**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** | **Email** |
| 1 | Đỗ Minh Nhật | 21522415 | 21522415@gm.uit.edu.vn |

1. **Phương thức làm việc:**
2. **Quy trình:**

* Chọn đề tài đồ án.
* Tìm hiểu tài liệu liên quan đến đề tài để thực hiện đồ án.
* Lên kế hoạch các mốc thời gian cho đồ án.
* Thực hiện đồ án, ra sản phẩm
* Kiểm tra, chỉnh sửa và hoàn thành đồ án.
* Đánh giá sản phẩm, hoàn thành báo cáo đồ án

1. **Công cụ:**

* Visual Studio 2022
* Visual Studio Code
* Youtube
* Facebook

1. **BÁO CÁO ĐỒ ÁN:**
2. **Giới thiệu chung: Đề tài đồ án “Chương trình Paint”**
3. **Giới thiệu đề tài:**

***Đề tài:*** Xây dựng chương trình Paint

* *Tên chương trình:* Demo Paint Version 1.0
* *Mục đích:* Áp dụng những kiến thức của môn Nhập môn lập trình C# (C# Programming Language) về lập trình trên môi trường Winform bằng ngôn ngữ C# và những kiến thức lập trình liên quan để xây dựng một ứng dụng hoàn chỉnh.
* *Yêu cầu:* Chương trình đáp ứng đủ các chức năng của một phần mềm vẽ trên máy tính. Ngoài ra, cung cấp thêm một số tính năng nâng cao cần thiết khác.

1. **Các chức năng của chương trình:**
2. **Chức năng cơ bản:**

* New, Open, Save,
* Cut, Copy, Paste
* Vẽ các đối tượng: Đường thằng, hình chữ nhật, hình ellipse, hình đa giác, hình cong, vẽ tự do...
* Tô màu cho các đối tượng.
* Chỉnh độ dày nét vẽ và màu sắc
* Lưu lại với các định dạng ảnh bitmap: bmp, jpg...

1. **Chức năng mới:**

* Cho phép các thao tác điều chỉnh đối tượng sau khi đã được vẽ:
* Di chuyển đối tượng
* Thay đổi kích thước đối tượng
* Thay đổi màu vẽ
* Xóa đối tượng
* Nhóm các đối tượng với nhau
* Chọn và xóa đối tượng.

1. **Phương pháp thực hiện:**
2. **Lập trình:**

* Kết hợp các kiến thức về: GDI+, WindowForm, Lập trình hướng đối tượng, Cấu trúc dữ liệu và giải thuật...

1. **Thiết kế giao diện:**

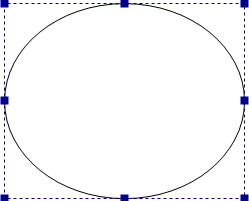
* Chương trình không chú trọng nhiều về trau chuốt phần giao diện, tuy nhiên giao diện vẫn được thiết kế và bố trí sao cho dễ hiểu và thuận tiện nhất cho người dùng.
* Lấy ý tưởng từ phần mềm Microsoft Paint, giao diện được thiết kế đơn giản nhưng thân thiện, dễ hiểu. Với hệ thống các phím tắt được thiết lập sẵn, ngoài thao tác xử lý bằng chuột người dùng có thể tương tác với phần mềm cả bằng bàn phím đối với một số chức năng.

1. **Ý tưởng thực hiện:**
2. **Mô hình đối tượng:**

* Tạo class DataManager để quản lý các biến sự kiện sẽ được thao tác trên giao diện Form và quản lý các biến dữ liệu linh động. Tạo biến shapeList để lưu lại tất cả các đối tượng sẽ được vẽ trên trang vẽ để thuận tiện cho việc chỉnh sửa đối tượng về sau.
* Tạo đối tượng Shape là lớp trừu tượng sẽ được các lớp hình khác kế thừa. Các thuộc tính và phương thức mặc định của lớp này là:
* *Thuộc tính:* tên hình vẽ (name), điểm đầu (pointHead), điểm cuối (pointTail), có đang được chọn hay không (isSelected), chiều dày nét vẽ (contourWidth), có đang ở chế độ đổ màu hay không (isFill), màu của nét vẽ (color), biến trừu tượng graphicsPath.
* *Phương thức:* Khởi tạo (Constructor), bản sao (Clone), vẽ hình (DrawShape), di chuyển hình (MoveShape), nhận biết điểm có nằm trong hình hay không (isHit), nhận biết hình có giao với hình chữ nhật cơ sở hay không (isCollideWithRegion), trả về hình chữ nhật cơ sở (GetRectangle), nhận biết điểm có đang ở gần điểm điều khiển không (isHitControlsPoint), thay đổi điểm điều khiển (ChangePoint), di chuyển điểm điều khiển (MoveControlPoint).

Điểm điều khiển

Điểm chuột click xuống trang vẽ



1. **Xây dựng chức năng:**
2. **Chức năng vẽ:**

* Sử dụng lệnh có sẵn trong thư viện GDI+ để vẽ các đối tượng hình cơ bản: đường thằng, hình chữ nhật, hình ellipse.
* Tính toán vị trí tương đối giữa các đỉnh của 1 đối tượng hình nâng cao: hình đa giác, hình cong...

1. **Chức năng di chuyển:**

* Xét vị trí tương đối của tọa độ chuột hiện hành và đối tượng hình, xác định có cho phép bật chức năng di chuyển không.
* Nếu tọa độ chuột hiện hành thuộc khu vực đối tượng chiếm giữ thì lấy tọa độ vị trí điểm chuột được nhấn và thả. Tính khoảng cách giữa 2 điểm đó và thực hiện tịnh tiến tọa độ các đỉnh của đối tượng theo khoảng cách tính được.

1. **Chức năng thay đổi kích thước:**

* Xét vị trí tương đối của tọa độ chuột hiện hành và đối tượng hình, xác định có cho phép bật chức năng di chuyển không.
* Nếu vị trí chuột trùng với tọa độ các điểm điều khiển thì xác định điểm điều khiển được chọn ở vị trí nào để tịnh tiến tọa độ các đỉnh của đối tượng cho phù hợp.

1. **Chức năng xóa:**

* Xét vị trí tương đối của tọa độ chuột hiện hành và đối tượng hình, xác định có cho phép bật chức năng xóa không.
* Nếu vị trí chuột nằm trong khu vực của đối tượng chiếm giữ thì lấy chỉ số đối tượng được chọn và thực hiện xóa đối tượng ra khỏi List danh sách đối tượng.

1. **Thiết kế dữ liệu:**
2. **Sơ đồ lớp đối tượng:**

HelpFunction

FindRegion

CurrentShapeStatus

PresenterUpdateImp

PresenterUpdate

PresenterDrawImp

PresenterDraw

PresenterAlterImp

PresenterAlter

Form1

ViewPaint

DataManager

MyPen

MyEllipse

Shape

GroupShape

MyPolygon

MyRectangle

MyLine

MyCurve

: Lớp đối tượng (class)

: Lớp đối tượng trừu tượng (abstract class)

: Giao diện (interface)

: Liệt kê (enum)

*Hình 1. Sơ đồ lớp đối tượng.*

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Tên lớp** | **Mô tả** | **Ghi chú** |
| 1 | Shape | Lớp đối tượng cha, xây dựng các thuộc tính, phương thức chung để các lớp đối tượng con kế thừa |  |
| 2 | MyCurve | Lớp hình cong |  |
| 3 | MyLine | Lớp đường thẳng |  |
| 4 | MyRectangle | Lớp hình chữ nhật |  |
| 5 | MyPolygon | Lớp hình đa giác |  |
| 6 | MyPen | Lớp hình vẽ tự do |  |
| 7 | MyEllipse | Lớp hình Ellipse |  |
| 8 | GroupShape | Lớp nhóm các hình vẽ |  |
| 9 | DataManager | Lớp quản lý dữ liệu linh động |  |
| 10 | ViewPaint | Interface các thao tác trên trang vẽ |  |
| 11 | PresenterAlter | Interface quản lý các thao tác về File |  |
| 12 | PresenterAlterImp | Triển khai của interface PresenterAlter |  |
| 13 | PresenterDraw | Interface quản lý các thao tác vẽ trên trang vẽ |  |
| 14 | PresenterDrawImp | Triển khai của interface PresenterDraw |  |
| 15 | PresenterUpdate | Interface quản lý các thao tác thay đổi thuộc tính vẽ |  |
| 16 | PresenterUpdateImp | Triển khai của interface PresenterUpdate |  |
| 17 | CurrentShapeStatus | Enum chứa các trạng thái của hình vẽ |  |
| 18 | FindRegion | Lớp thực hiện các thao tác đối với hình chữ nhật cơ sở |  |
| 19 | HelpFunction | Lớp chứa các hàm hỗ trợ khác |  |
| 20 | Form1 | Form thiết kế chính của chương trình |  |

*Bảng 1. Các lớp đối tượng của chương trình.*

1. **Xây dựng các lớp:**
2. ***Lớp Shape:***

Shape

* Lớp *Shape* là một lớp triển khai của interface *Icloneable*.
* Đây là lớp đối tượng trừu tượng, xây dựng các thuộc tính, phương thức chung để các lớp đối tượng con kế thừa.

|  |  |  |  |
| --- | --- | --- | --- |
| Shape | | | |
|  | **Tên** | **Kiểu dữ liệu** | **Ý nghĩa** |
| **Thuộc tính** | name | string | Tên của hình |
| pointHead | Point | Điểm đầu của hình chữ nhật cơ sở |
| pointTail | Point | Điểm cuối của hình chữ nhật cơ sở |
| isSelected | bool | Cho biết hình hiện tại có được chọn hay không |
| contourWidth | int | Độ dày của nét vẽ |
| isFill | bool | Cho biết hình có đang ở chế độ Fill hay không |
| color | Color | Màu của nét vẽ |
| graphicsPath | GraphicsPath | Tạo ra đối tượng *graphicsPath* |
|  | **Tên** | **Kiểu trả về** | **Ý nghĩa** |
| **Phương thức** | Clone() | object | Tạo ra một đối tượng copy từ đối tượng này |
| DrawShape(Graphics g) | void | Vẽ hình hiện tại lên graphics |
| MoveShape(Point distance) | void | Di chuyển hình hiện tại với khoảng cách distance |
| isHit(Point p) | bool | Cho biết điểm p có thuộc hình này hay không |
| isCollideWithRegion(Rectangle rectangle) | bool | Cho biết hình chữ nhật rectangle có giao với hình này hay không |
| GetRectangle() | Rectangle | Nhận hình chữ nhật cơ sở |
| isHitControlsPointPoint(Point p) | int | Cho biết điểm p có phải là điểm điều khiển hay không |
| ChangePoint(int index) | void | Thay đổi lại điểm đầu và điểm cuối của hình vẽ theo điểm điều khiển |
| MoveControlPoint(Point current, Point previous, int index) | void | Điều chỉnh hình theo các điểm điều khiển |

*Bảng 2. Lớp trừu tượng Shape.*

1. ***Lớp MyRectangle:***

MyRectangle

Shape

* Các thuộc tính trong lớp *MyRectangle* được kế thừa từ lớp *Shape*
* Override lại các phương thức từ lớp *Shape*.

|  |  |  |  |
| --- | --- | --- | --- |
| MyRectangle | | | |
|  | **Tên** | **Kiểu dữ liệu** | **Ý nghĩa** |
| **Thuộc tính** | graphicsPath | GraphicsPath | Tạo ra đối tượng *graphicsPath* |
|  | **Tên** | **Kiểu trả về** | **Ý nghĩa** |
| **Phương thức** | MyRectangle() |  | Constructor không tham số của lớp MyRectangle |
| Clone() | object | Tạo ra một đối tượng MyRectangle copy từ đối tượng này |
| DrawShape(Graphics g) | void | Vẽ hình chữ nhật hiện tại lên graphics |
| isHit(Point p) | bool | Cho biết điểm p có thuộc hình chữ nhật này hay không |

*Bảng 3. Lớp MyRectangle.*

1. ***Lớp MyLine:***

MyRectangle

Shape

MyLine

* Các thuộc tính trong lớp *MyLine* được kế thừa từ lớp *Shape*
* Override lại các phương thức từ lớp *Shape*.

|  |  |  |  |
| --- | --- | --- | --- |
| MyLine | | | |
|  | **Tên** | **Kiểu dữ liệu** | **Ý nghĩa** |
| **Thuộc tính** | graphicsPath | GraphicsPath | Tạo ra đối tượng GraphicsPath |
|  | **Tên** | **Kiểu trả về** | **Ý nghĩa** |
| **Phương thức** | MyLine() |  | *Constructor* không tham số của lớp *MyLine*. |
| Clone() | object | Tạo ra một đối tượng *MyLine* copy từ đối tượng này. |
| DrawShape(Graphics g) | void | Vẽ đường thẳng hiện tại lên *graphics*. |
| isHit(Point p) | bool | Cho biết điểm p có thuộc đường thẳng này hay không. |
| isHitControlsPoint(Point p) | int | Cho biết điểm p có phải là điểm điều khiển của đường thẳng hay không. |

*Bảng 4. Lớp MyLine.*

1. ***Lớp MyPolygon***

MyRectangle

Shape

MyLine

MyPolygon

* Các thuộc tính trong lớp *MyPolygon* được kế thừa từ lớp *Shape*.
* Override lại các phương thức từ lớp *Shape*.

|  |  |  |  |
| --- | --- | --- | --- |
| MyPolygon | | | |
|  | **Tên** | **Kiểu dữ liệu** | **Ý nghĩa** |
| **Thuộc tính** | points | List<Point> | Danh sách các điểm |
| graphicsPath | GraphicsPath | Tạo ra đối tượng *graphicsPath* |
|  | **Tên** | **Kiểu trả về** | **Ý nghĩa** |
| **Phương thức** | MyPolygon() |  | *Constructor* không tham số của lớp *MyPolygon* |
| Clone() | object | Tạo ra một đối tượng *MyPolygon* copy từ đối tượng này |
| DrawShape(Graphics g) | void | Vẽ hình đa giác hiện tại lên *graphics* |
| isHit(Point p) | bool | Cho biết điểm p có thuộc đa giác này hay không |
| MoveShape(Point distance) | void | Di chuyển hình đa giác với khoảng cách distance. |
| isHitControlsPoint(Point p) | int | Cho biết điểm p có phải là điểm điều khiển của đa giác hay không |
| MoveControlPoint(Point pointCurrent, Point previous, int index) | void | Điều chỉnh đa giác theo các điểm điều khiển. |

*Bảng 5. Lớp MyPolygon.*

1. ***Lớp MyCurve:***

MyCurve

MyLine

MyPolygon

MyRectangle

Shape

* Các thuộc tính trong lớp *MyCurve* được kế thừa từ lớp *Shape*
* Override lại các phương thức từ lớp *Shape*.

|  |  |  |  |
| --- | --- | --- | --- |
| * MyCurve | | | |
|  | **Tên** | **Kiểu dữ liệu** | **Ý nghĩa** |
| **Thuộc tính** | points | List<Point> | Danh sách các điểm |
| graphicsPath | GraphicsPath | Tạo ra đối tượng *graphicsPath* |
|  | **Tên** | **Kiểu trả về** | **Ý nghĩa** |
| **Phương thức** | MyCurve() |  | *Constructor* không tham số của lớp *MyCurve* |
| Clone() | object | Tạo ra một đối tượng *MyCurve* copy từ đối tượng này |
| DrawShape(Graphics g) | void | Vẽ hình đa giác hiện tại lên *graphics* |
| isHit(Point p) | bool | Cho biết điểm p có thuộc hình cong này hay không |
| MoveShape(Point distance) | void | Di chuyển hình cong với khoảng cách distance. |
| isHitControlsPoint(Point p) | int | Cho biết điểm p có phải là điểm điều khiển của hình cong hay không |
| MoveControlPoint(Point pointCurrent, Point previous, int index) | void | Điều chỉnh hình cong theo các điểm điều khiển. |

*Bảng 6. Lớp MyCurve.*

1. ***Lớp MyEllipse:***

MyCurve

MyLine

MyPolygon

MyRectangle

Shape

MyEllipse

* Các thuộc tính trong lớp *MyEllipse* được kế thừa từ lớp *MyRectangle*
* Override lại các phương thức từ lớp *MyRectangle*.

|  |  |  |  |
| --- | --- | --- | --- |
| MyEllipse | | | |
|  | **Tên** | **Kiểu dữ liệu** | **Ý nghĩa** |
| **Thuộc tính** | graphicsPath | GraphicsPath | Tạo ra đối tượng graphicsPath |
|  | **Tên** | **Kiểu trả về** | **Ý nghĩa** |
| **Phương thức** | MyEllipse() |  | *Constructor* không tham số của lớp *MyEllipse*. |
| Clone() | object | Tạo ra một đối tượng *MyEllipse* copy từ đối tượng này. |
| graphicsPath | GraphicsPath | Tạo ra đối tượng *graphicsPath.* |

*Bảng 7. Lớp MyEllipse.*

1. ***Lớp MyPen:***

MyCurve

MyLine

MyPolygon

MyRectangle

Shape

MyPen

MyEllipse

* Các thuộc tính trong lớp *MyPen* được kế thừa từ lớp *MyCurve.*
* Override lại các phương thức từ lớp *MyCurve*.

|  |  |  |  |
| --- | --- | --- | --- |
| MyEllipse | | | |
|  | **Tên** | **Kiểu dữ liệu** | **Ý nghĩa** |
| **Thuộc tính** | isEraser | bool | Cho biết thuộc tính của MyPen là bút hay là tẩy. |
|  | **Tên** | **Kiểu trả về** | **Ý nghĩa** |
| **Phương thức** | MyPen() |  | *Constructor* không tham số của lớp *MyPen*. |

*Bảng 8. Lớp MyPen.*

1. ***Lớp GroupShape:***

MyCurve

MyLine

MyPolygon

MyRectangle

Shape

GroupShape

MyPen

MyEllipse

* Các thuộc tính trong lớp *GroupShape* được kế thừa từ lớp *Shape.*
* Override lại các phương thức từ lớp *Shape*.

|  |  |  |  |
| --- | --- | --- | --- |
| MyEllipse | | | |
|  | **Tên** | **Kiểu dữ liệu** | **Ý nghĩa** |
| **Thuộc tính** | shapes | List<Shape> | Danh sách các hình. |
| this[int index] | Shape | Lấy hình từ vị trí index. |
| graphicsPaths | GraphicsPath[] | Tạo ra mảng các đối tượng GraphicPath |
| graphicsPath | GraphicsPath | Tạo ra đối tượng graphicsPath |
| Count | int | Số lượng các hình trong group |
|  | **Tên** | **Kiểu trả về** | **Ý nghĩa** |
| **Phương thức** | GroupShape() |  | *Constructor* không tham số của lớp *GroupShape*. |
| AddShape(Shape shape) | void | Thêm một hình vào danh sách. |
| Clone() | object | Tạo ra một đối tượng *GroupShape* copy từ đối tượng này. |
| DrawShape(Graphics g) | void | Vẽ các hình trong danh sách lên graphics |
| isHit(Point p) | bool | Cho biết điểm p có thuộc group này hay không |
| GetEnumerator() | IEnumerator | Lấy enumerator của shapes hiện tại |
| MoveShape(Point distance) | void | Di chuyển group với khoảng cách distance |

*Bảng 9. Lớp GroupShape.*

1. ***Lớp DataManager:***

DataManager

MyCurve

MyLine

MyPolygon

MyRectangle

Shape

GroupShape

MyPen

MyEllipse

* Lớp *DataManager* quản lý các đối tượng hỗ trợ để vẽ hình.

|  |  |  |  |
| --- | --- | --- | --- |
| DataManager | | | |
|  | **Tên** | **Kiểu dữ liệu** | **Ý nghĩa** |
| **Thuộc tính** | instance | DataManager | Instance của lớp DataManager. |
| shapeList | List<Shape> | Chứa danh sách các hình vẽ. |
| savedShapes | List<Shape> | Chứa danh sách các hình vẽ trong bộ nhớ tạm. |
| shapeToMove | Shape | Chứa hình được chọn hiện tại. |
| rectangleRegion | Rectangle | Vẽ vùng được chọn có kích thước như đối tượng này. |
| isMouseDown | bool | Thiết lập trạng thái click chuột xuống của người dùng. |
| isMovingShape | bool | Thiết lập trạng thái di chuyển hình của người dùng. |
| isMovingMouse | bool | Thiết lập trạng thái di chuyển chuột của người dùng. |
| isDrawingCurve | bool | Thiết lập trạng thái cho biết có phải người dùng đang vẽ đường cong hay không |
| isDrawingPolygon | bool | Thiết lập trạng thái cho biết có phải người dùng đang vẽ đa giác hay không. |
| isDrawingPen | bool | Thiết lập trạng thái cho biết có phải người dùng đang vẽ pencil hay không. |
| isDrawingEraser | bool | Thiết lập trạng thái cho biết có phải người dùng đang chọn chế độ eraser hay không. |
| isFill | bool | Thiết lập trạng thái cho biết có phải người dùng đang chọn chế độ fill hay không. |
| isSave | bool | Thiết lập trạng thái cho biết có phải người dùng đã lưu hình hay chưa. |
| isNotNone | bool | Thiết lập trạng thái cho biết có background hiện tại được vẽ hay là chưa. |
| pointToResize | int | Cho biết điểm điều khiển nào người dùng muốn resize kích thước. |
| currentShape | CurrentShapeStatus | Cho biết trạng thái hình hiện tại là hình nào. |
| cursorCurrent | Point | Cho biết vị trí con trỏ chuột hiện tại. |
| colorCurrent | Color | Cho biết màu của nét vẽ hiện tại. |
| lineSize | int | Cho biết độ dày của nét vẽ. |
|  | **Tên** | **Kiểu trả về** | **Ý nghĩa** |
| **Phương thức** | DataManager() |  | *Constructor* không tham số của lớp *DataManager*. |
| getInstance() | DataManager | Nhận instance của lớp *DataManager*. |
| UpdatePointTail(Point p) | void | Cập nhật điểm cuối của hình vẽ. |
| AddEntity(Shape shape) | void | Thêm đối tượng vào danh sách *shapeList*. |
| AddSavedShapes(Shape shape) | void | Thêm đối tượng vào danh sách *savedShapes*. |
| offAllShapeSelected() | void | Đặt trạng thái được chọn của tất cả các hình về false. |
| distanceXY(Point x, Point y) | Point | Tính khoảng cách giữa điểm x và y. |
| UpdateRectangleRegion(Point p) | void | Cập nhật lại một vùng là hình chữ nhật bao quanh hình vẽ. |

*Bảng 10. Lớp DataManager.*

1. ***Interface PresenterAlter và lớp PresenterAlterImp:***

DataManager

MyCurve

MyLine

MyPolygon

MyRectangle

Shape

GroupShape

MyPen

MyEllipse

PresenterAlterImp

PresenterAlter

* Lớp *PresenterAlterImp* là lớp triển khai của interface *PresenterAlter.*

|  |  |  |  |
| --- | --- | --- | --- |
| PresenterAlter | | | |
|  | **Tên** | **Kiểu trả về** | **Ý nghĩa** |
| **Phương thức** | onClickDrawGroup() | void | Phương thức group một hình. |
| onClickDrawUngroup() | void | Phương thức ungroup một hình. |
| onClickDeleteShape() | void | Phương thức xóa một hình. |
| onClickCopyShape() | void | Phương thức copy một hình. |
| onClickCutShape() | void | Phương thức cut một hình. |
| onClickPasteShape() | void | Phương thức paste một hình. |
| onClickClearAll(PictureBox pictureBox) | void | Phương thức xóa tất cả các hình. |
| onClickSaveImage(PictureBox pictureBox) | void | Phương thức lưu một hình |
| onClickOpenImage(PictureBox pictureBox) | void | Phương thức mở một hình lên pictureBox |
| onClickNewImage(PictureBox pictureBox) | void | Phương thức tạo một hình vẽ mới từ pictureBox |
| onUseKeyStrokes(PictureBox pictureBox) | void | Phương thức xử lý nhất tổ hợp phím trên pictureBox |

*Bảng 11. Interface PresenterAlter.*

|  |  |  |  |
| --- | --- | --- | --- |
| PresenterAlterImp | | | |
|  | **Tên** | **Kiểu dữ liệu** | **Ý nghĩa** |
| **Thuộc tính** | viewPaint | ViewPaint | Tạo ra đối tượng *ViewPaint*. |
| dataManager | DataManager | Tạo ra đối tượng *DataManager*. |
|  | **Tên** | **Kiểu trả về** | **Ý nghĩa** |
| **Phương thức** | PresenterAlterImp(ViewPaint viewPaint) |  | *Constructor* có tham số của lớp *PresenterAlterImp*. |
| onClickDrawGroup() | void | Phương thức group một hình. |
| onClickDrawUngroup() | void | Phương thức ungroup một hình. |
| onClickDeleteShape() | void | Phương thức xóa một hình. |
| onClickCopyShape() | void | Phương thức copy một hình. |
| onClickCutShape() | void | Phương thức cut một hình. |
| onClickPasteShape() | void | Phương thức paste một hình. |
| onClickClearAll(PictureBox pictureBox) | void | Phương thức xóa tất cả các hình. |
| onClickSaveImage(PictureBox pictureBox) | void | Phương thức lưu một hình |
| onClickOpenImage(PictureBox pictureBox) | void | Phương thức mở một hình lên pictureBox |
| onClickNewImage(PictureBox pictureBox) | void | Phương thức tạo một hình vẽ mới từ pictureBox |
| onUseKeyStrokes(PictureBox pictureBox) | void | Phương thức xử lý nhất tổ hợp phím trên pictureBox |

*Bảng 12. Lớp PresenterAlterImp.*

1. ***Interface PresenterAlter và lớp PresenterAlterImp:***

DataManager

MyCurve

MyLine

MyPolygon

MyRectangle

Shape

GroupShape

MyPen

MyEllipse

PresenterAlterImp

PresenterAlter

PresenterDrawImp

PresenterDraw

* Lớp *PresenterDrawImp* là lớp triển khai của interface *PresenterDraw.*

|  |  |  |  |
| --- | --- | --- | --- |
| PresenterDraw | | | |
|  | **Tên** | **Kiểu trả về** | **Ý nghĩa** |
| **Phương thức** | GetDrawing(Graphics g) | void | Phương thức vẽ một hình lên graphics g. |
| onClickMouseDown(Point p) | void | Phương thức xử lý hạ chuột từ người dùng. |
| onClickMouseMove(Point p) | void | Phương thức xử lý di chuyển chuột từ người dùng. |
| onClickMouseUp() | void | Phương thức xử lý thả chuột từ người dùng. |
| onClickDrawLine() | void | Phương thức gọi vẽ đường thẳng. |
| onClickDrawRectangle() | void | Phương thức gọi vẽ hình chữ nhật. |
| onClickDrawEllipse() | void | Phương thức gọi vẽ hình ellipse. |
| onClickDrawBezier() | void | Phương thức gọi vẽ đường cong. |
| onClickDrawPolygon() | void | Phương thức gọi vẽ đa giác. |
| onClickDrawPen() | void | Phương thức gọi vẽ tự do. |
| onClickDrawEraser() | void | Phương thức gọi vẽ xóa. |
| onClickStopDrawing(MouseButtons mouse) | void | Phương thức xử lý chuột phải từ người dùng. |

*Bảng 13. Interface PresenterDraw.*

|  |  |  |  |
| --- | --- | --- | --- |
| PresenterDrawImp | | | |
|  | **Tên** | **Kiểu dữ liệu** | **Ý nghĩa** |
| **Thuộc tính** | viewPaint | ViewPaint | Tạo ra đối tượng *ViewPaint*. |
| dataManager | DataManager | Tạo ra đối tượng *DataManager*. |
|  | **Tên** | **Kiểu trả về** | **Ý nghĩa** |
| **Phương thức** | PresenterDrawImp(ViewPaint viewPaint) |  | *Constructor* có tham số của lớp *PresenterDrawImp*. |
| GetDrawing(Graphics g) |  | Phương thức vẽ một hình lên graphics g. |
| onClickMouseDown(Point p) | void | Phương thức xử lý hạ chuột từ người dùng. |
| onClickMouseMove(Point p) | void | Phương thức xử lý di chuyển chuột từ người dùng. |
| onClickMouseUp() | void | Phương thức xử lý thả chuột từ người dùng. |
| onClickDrawLine() | void | Phương thức gọi vẽ đường thẳng. |
| onClickDrawRectangle() | void | Phương thức gọi vẽ hình chữ nhật. |
| onClickDrawEllipse() | void | Phương thức gọi vẽ hình ellipse. |
| onClickDrawBezier() | void | Phương thức gọi vẽ đường cong. |
| onClickDrawPolygon() | void | Phương thức gọi vẽ đa giác. |
| onClickDrawPen() | void | Phương thức gọi vẽ tự do. |
| onClickDrawEraser() | void | Phương thức gọi vẽ xóa. |
| onClickStopDrawing(MouseButtons mouse) | void | Phương thức xử lý chuột phải từ người dùng. |
| handleClickToSelect(Point p) | void | Phương thức quản lý trường hợp click để chọn. |
| handleClickToDraw(Point p) | void | Phương thức quản lý trường hợp click để vẽ. |
| DrawRegionForShape(Shape shape, Graphics g) | void | Phương thức gọi vẽ hình chữ nhật cơ sở cho hình. |
| SetDefaultToDraw() | void | Phương thức thiết lập các thông số mặc định trước khi vẽ. |

*Bảng 14. Lớp PresenterDrawImp.*

1. ***Interface PresenterUpdate và lớp PresenterUpdateImp:***

DataManager

MyCurve

MyLine

MyPolygon

MyRectangle

Shape

GroupShape

MyPen

MyEllipse

PresenterAlterImp

PresenterAlter

PresenterDrawImp

PresenterDraw

PresenterUpdateImp

PresenterUpdate

* Lớp *PresenterUpdateImp* là lớp triển khai của interface *PresenterUpdate.*

|  |  |  |  |
| --- | --- | --- | --- |
| PresenterUpdate | | | |
|  | **Tên** | **Kiểu trả về** | **Ý nghĩa** |
| **Phương thức** | onClickSelectMode() | void | Phương thức xử lý khi người dùng chọn chế độ select. |
| onClickSelectColor(Color color, Graphics g) | void | Phương thức xử lý khi người dùng chọn thay đổi màu sắc. |
| onClickSelectSize(int size) | void | Phương thức xử lý khi người dùng chọn thay đổi kích thước đường vẽ. |
| onClickSelectFill(Button btn, Graphics g) | void | Phương thức xử lý khi người dùng chọn chế độ fill. |

*Bảng 15. Interface PresenterUpdate.*

|  |  |  |  |
| --- | --- | --- | --- |
| PresenterUpdateImp | | | |
|  | **Tên** | **Kiểu dữ liệu** | **Ý nghĩa** |
| **Thuộc tính** | viewPaint | ViewPaint | Tạo ra đối tượng *ViewPaint*. |
| dataManager | DataManager | Tạo ra đối tượng *DataManager*. |
|  | **Tên** | **Kiểu trả về** | **Ý nghĩa** |
| **Phương thức** | PresenterUpdateImp(ViewPaint viewPaint) |  | *Constructor* có tham số của lớp *PresenterUpdateImp*. |
| onClickSelectMode() | void | Phương thức xử lý khi người dùng chọn chế độ select. |
| onClickSelectColor(Color color, Graphics g) | void | Phương thức xử lý khi người dùng chọn thay đổi màu sắc. |
| onClickSelectSize(int size) | void | Phương thức xử lý khi người dùng chọn thay đổi kích thước đường vẽ. |
| onClickSelectFill(Button btn, Graphics g) | void | Phương thức xử lý khi người dùng chọn chế độ fill. |

*Bảng 16. Lớp PresenterUpdateImp.*

1. ***Interface ViewPaint:***

DataManager

MyCurve

MyLine

MyPolygon

MyRectangle

Shape

GroupShape

MyPen

MyEllipse

PresenterAlterImp

PresenterAlter

PresenterDrawImp

PresenterDraw

PresenterUpdateImp

PresenterUpdate

ViewPaint

* Interface *ViewPaint* là giao diện kế thừa của form hiển thị*.*

|  |  |  |  |
| --- | --- | --- | --- |
| ViewPaint | | | |
|  | **Tên** | **Kiểu trả về** | **Ý nghĩa** |
| **Phương thức** | RefreshDrawing() | void | Phương thức gọi vẽ lại bản vẽ. |
| SetCursor(Cursor cursor) | void | Phương thức thiết lập hình dạng cho con trỏ chuột. |
| SetColor(Color color) | void | Phương thức thiết lập màu sắc cho background. |
| SetColor(Button btn, Color color) | void | Phương thức thiết lập hình nền cho button. |
| SetDrawing(Shape shape, Graphics g) | void | Phương thức vẽ một hình lên graphics. |
| SetDrawingLineSelected(Shape shape, Brush brush, Graphics g) | void | Phương thức vẽ điểm điều khiển cho đường thẳng. |
| SetDrawingCurveSelected(List<Point> points, Brush brush, Graphics g) | void | Phương thức vẽ điểm điều khiển cho đường cong. |
| SetDrawingRegionRectangle(Pen p, Rectangle rectangle, Graphics g) | void | Phương thức vẽ điểm điều khiẻn cho hình vẽ bởi pen. |
| MovingShape(Shape shape, Point distance) | void | Phương thức di chuyển một hình. |
| MovingControlPoint(Shape shape, Point pointCurrent, Point previous, int indexPoint) | void | Phương thức điều chỉnh một hình theo điểm điều khiển. |

*Bảng 15. Interface ViewPaint.*

1. ***Enum CurrentShapeStatus:***

DataManager

MyCurve

MyLine

MyPolygon

MyRectangle

Shape

GroupShape

MyPen

MyEllipse

CurrentShapeStatus

PresenterAlterImp

PresenterAlter

PresenterDrawImp

PresenterDraw

PresenterUpdateImp

PresenterUpdate

ViewPaint

* Enum *CurrentShapeStatus* chứa các trạng thái của hình vẽ*.*

|  |  |  |
| --- | --- | --- |
| CurrentShapeStatus | | |
|  | **Tên** | **Ý nghĩa** |
| **Phần tử** | Void | Đại diện cho thuộc tính chưa được chọn. |
| Line | Đại diện cho thuộc tính đường thẳng. |
| Rectangle | Đại diện cho thuộc tính hình chữ nhật. |
| Ellipse | Đại diện cho thuộc tính hình ellipse. |
| Square | Đại diện cho thuộc tính hình vuông. |
| Circle | Đại diện cho thuộc tính hình tròn. |
| Curve | Đại diện cho thuộc tính đường cong. |
| Polygon | Đại diện cho thuộc tính đa giác. |
| Pen | Đại diện cho thuộc tính vẽ tự do. |
| Eraser | Đại diện cho thuộc tính vẽ xóa. |

*Bảng 16. Enum CurrentShapeStatus.*

1. ***Lớp FindRegion:***

DataManager

MyCurve

MyLine

MyPolygon

MyRectangle

Shape

GroupShape

MyPen

MyEllipse

CurrentShapeStatus

PresenterAlterImp

PresenterAlter

PresenterDrawImp

PresenterDraw

FindRegion

PresenterUpdateImp

PresenterUpdate

ViewPaint

* Lớp *FindRegion* dùng để thao tác với vùng bao bọc của hình vẽ.

|  |  |  |  |
| --- | --- | --- | --- |
| FindRegion | | | |
|  | **Tên** | **Kiểu trả về** | **Ý nghĩa** |
| **Phương thức** | SetPointHeadTail(GroupShape group) | void | Phương thức thiết lập điểm đầu và điểm cuối cho hình thuộc lớp *GroupShape*. |
| SetPointHeadTail(MyCurve curve) | void | Phương thức thiết lập điểm đầu và điểm cuối cho hình thuộc lớp *MyCurve*. |
| SetPointHeadTail(MyPolygon polygon) | void | Phương thức thiết lập điểm đầu và điểm cuối cho hình thuộc lớp *MyPolygon*. |
| SetPointHeadTail(MyPen pen) | void | Phương thức thiết lập điểm đầu và điểm cuối cho hình thuộc lớp *MyPen*. |
| GetControlPoints(Shape shape) | List<Point> | Phương thức nhật tất cả điểm điều khiển của một hình. |

*Bảng 17. Lớp FindRegion.*

1. ***Lớp HelpFunction:***

DataManager

MyCurve

MyLine

MyPolygon

MyRectangle

Shape

GroupShape

MyPen

MyEllipse

CurrentShapeStatus

PresenterAlterImp

PresenterAlter

PresenterDrawImp

PresenterDraw

FindRegion

PresenterUpdateImp

PresenterUpdate

HelpFunction

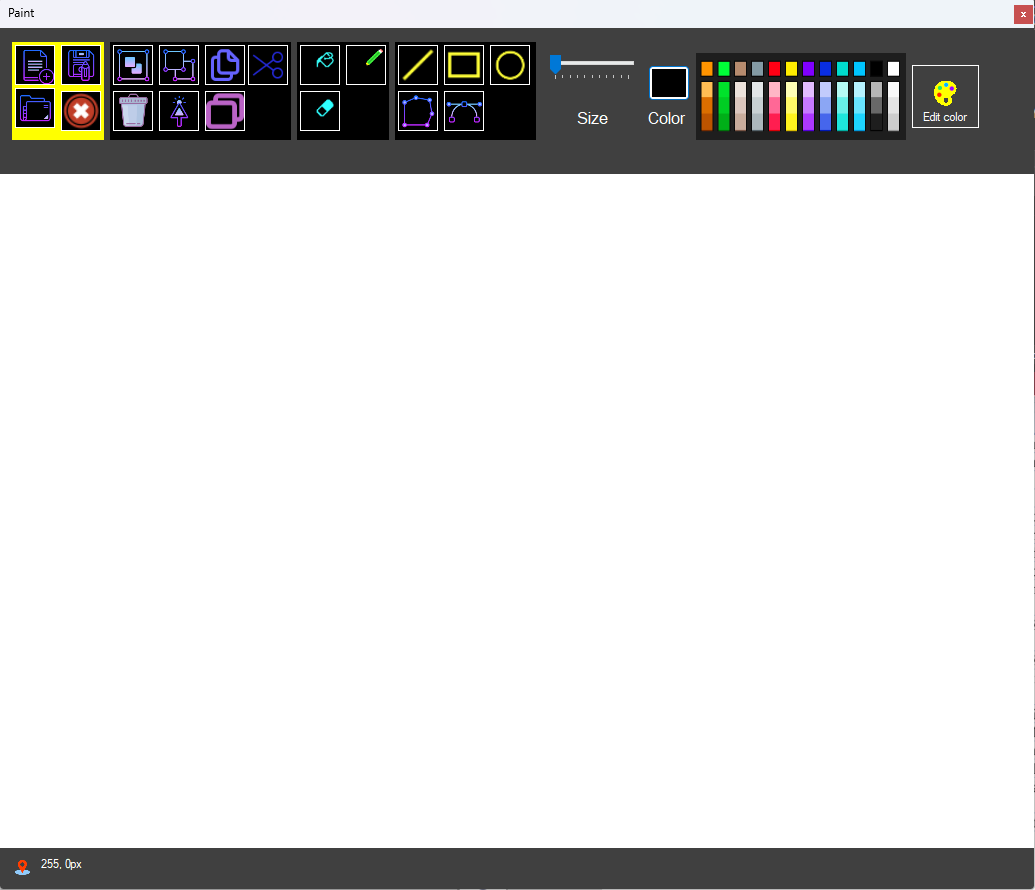
ViewPaint

* Lớp *HelpFunction* cung cấp các hàm hỗ trợ khác.

|  |  |  |  |
| --- | --- | --- | --- |
| HelpFunction | | | |
|  | **Tên** | **Kiểu trả về** | **Ý nghĩa** |
| **Phương thức** | SetPoint(PictureBox pb, Point pt) | Point | Phương thức lấy điểm trên picturebox pb tỉ lệ với điểm pt. |
| GetRectangle(Point a, Point b) | Rectangle | Phương thức lấy hình chữ nhật từ 2 điểm a và b |
| isInside(Shape shape, Point p) | bool | Phương thức kiểm tra điểm p có nằm trong hình shape hay không. |

*Bảng 17. Lớp HelpFunction.*

1. **Thiết kế giao diện:**

****

*Hình 2. Giao diện chương trình Demo Paint.*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Tên** | **Kiểu** | **Chức năng** | **Chi chú** |
| 1 | New | Button | Tạo trang vẽ mới. |  |
| 2 | Open | Button | Mở file ảnh có sẵn trong máy tính. |  |
| 3 | Save | Button | Lưu trang vẽ thành file ảnh. |  |
| 4 | Clear | Button | Xóa toàn bộ hình vẽ trên trang vẽ. |  |
| 5 | Group shape | Button | Nhóm các hình được chọn. |  |
| 6 | Ungroup shape | Button | Rã nhóm nhóm được chọn. |  |
| 7 | Delete | Button | Xóa hình được chọn. |  |
| 8 | Select | Button | Chuyển sang chế độ chọn hình. |  |
| 9 | Copy | Button | Copy hình được chọn vào bộ nhớ tạm. |  |
| 10 | Cut | Button | Cut hình được chọn và lưu vào bộ nhớ tạm. |  |
| 11 | Paste | Button | Paste các hình trong bộ nhớ tạm ra trang vẽ. |  |
| 12 | Fill | Button | Chuyển sang chế độ Fill. |  |
| 13 | Eraser | Button | Chuyển sang chế độ vẽ xóa. |  |
| 14 | Pencil | Button | Chuyển sang chế độ vẽ tự do |  |
| 15 | Draw Line | Button | Chuyển sang chế độ vẽ đường thẳng. |  |
| 16 | Draw Rectangle | Button | Chuyển sang chế độ vẽ hình chữ nhật. |  |
| 17 | Draw Ellipse | Button | Chuyển sang chế độ vẽ hình ellipse. |  |
| 18 | Draw Polygon | Button | Chuyển sang chế độ vẽ hình đa giác. |  |
| 19 | Line Size | TrackBar | Điều chỉnh độ dày nét vẽ. |  |
| 20 | Pick Color | Button | Hiển thị màu được chọn hiện tại. |  |
| 21 | Color Picker | PictureBox | Bảng chọn màu vẽ. |  |
| 22 | Edit Color | Button | Điều chỉnh màu vẽ. |  |
| 23 | Location | Label | Hiển thị tọa độ của con trỏ chuột hiện tại. |  |

*Bảng 18. Các đối tượng của giao diện chương trình.*

# **TÀI LIỆU THAM KHẢO**

[1] Source code tham khảo: <http://doc.edu.vn/tai-lieu/de-tai-xay-dung-chuong-trinh-ve-hinh-giong-microsoft-paint-54976/>

[2] Hướng dẫn sử dụng thư viện GDI+: <https://msdn.microsoft.com/en-us/library/ms533798(v=vs.85).aspx>

**-------------------- HẾT ---------------------**