**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



**BÁO CÁO ĐỒ ÁN**

**ĐỒ ÁN 1**

**VIẾT ỨNG DỤNG VẼ VÀ QUẢN LÍ TỰA ACAD**

**SVTH1 : BÙI THỊ HỒNG NHUNG – 17110201**

**SVTH2 : NGUYỄN TẤN ĐẠT – 17110216**

**GVHD : NGUYỄN ĐĂNG QUANG**

**TP. Hồ Chí Minh, tháng 11 năm 2019**

**MỤC LỤC**

1. [DANH MỤC HÌNH ẢNH 4](#_Toc23978031)
2. [DANH MỤC BẢNG 4](#_Toc23978032)
3. [Chương I: Phần mở đầu 5](#_Toc23978033)

[1. Giới thiệu 5](#_Toc23978034)

[2. Yêu cầu phần mềm 5](#_Toc23978035)

[3. Mục tiêu 5](#_Toc23978036)

1. [Chương II : Nội dung 6](#_Toc23978037)

[1. Phần mềm dùng để làm gì? 6](#_Toc23978038)

[2. Dữ liệu, thông tin đầu vào 6](#_Toc23978039)

[3. Mục đích, tính năng của ứng dụng 6](#_Toc23978040)

[4. Giao diện dự kiến 7](#_Toc23978041)

1. [CHƯƠNG III : PHÂN CÔNG 8](#_Toc23978042)
2. [CHƯƠNG IV : THIẾT KẾ 9](#_Toc23978043)

[1. Thiết kế lớp 9](#_Toc23978044)

[*1.1* *Danh mục các lớp* 10](#_Toc23978045)

[*Bảng 3. Danh mục các lớp* 10](#_Toc23978046)

[*1.2* *Đặc tả phương thức trong các lớp* 11](#_Toc23978047)

[*1.2.1* *Lớp Interface (iShape)* 11](#_Toc23978048)

[*Bảng 4. Phương thức của interface iShape* 11](#_Toc23978049)

[*1.2.2* *Lớp Abstract (Shape)* 12](#_Toc23978050)

[*Bảng 5. Phương thức của lớp Abstract Shape* 12](#_Toc23978051)

[1.2.3 Lớp MainSence 13](#_Toc23978052)

[1.2.4 Lớp Triangle 16](#_Toc23978053)

[*1.2.5* *Lớp circularArc* 16](#_Toc23978054)

[*1.2.6* *Lớp Polyline* 17](#_Toc23978055)

[2. Thiết kế giao diện 18](#_Toc23978056)

1. [CHƯƠNG V : CÀI ĐẶT VÀ KIỂM THỬ 18](#_Toc23978057)
2. [CHƯƠNG VI : KẾT LUẬN 19](#_Toc23978058)

[1. Kết quả đạt được 20](#_Toc23978059)

[2. Kết luận 20](#_Toc23978060)

[*a.* *Ưu điểm* 20](#_Toc23978061)

[*b.* *Nhược điểm* 20](#_Toc23978062)

[3. Đề xuất phương án cải thiện 20](#_Toc23978063)

1. [PHỤ LỤC 20](#_Toc23978064)

[TÀI LIỆU THAM KHẢO 21](#_Toc23978065)

DANH MỤC HÌNH ẢNH

*Hình 1. Giao diện chương trình dự kiến*

*Hình 2. Class Diagram*

*Hình 3. Giao diện chương trình*

*Hình 4. Giao diện hiển thị các hình được vẽ*

DANH MỤC BẢNG

*Bảng 1. Các Control được sử dụng trong chương trình*

*Bảng 2. Bảng phân công công việc*

*Bảng 3. Danh mục các lớp*

*Bảng 4. Phương thức của interface iShape*

*Bảng 5. Phương thức của lớp Abstract Shape*

*Bảng 6. Phương thức của lớp MainSence*

*Bảng 7. Phương thức của lớp Triangle*

*Bảng 8. Phương thức của lớp circularArc*

*Bảng 9. Phương thức của lớp Polyline*

*Bảng 10. Giao diện của chương trình*

*Bảng 11. Kết quả xây dựng chương trình*

Chương I: Phần mở đầu

1. Giới thiệu

Chương trình được xây dựng nhằm củng cố kiến thức đã học ở các môn trước, lập trình hướng đối tượng và tìm hiểu về ngôn ngữ mới Java. Mục đích của đề tài là tạo ra một chương trình tựa AutoCad, vẽ các hình học căn bản thông qua ngôn ngữ Java và tính chất của một đối tượng.

1. Yêu cầu phần mềm

Xây dựng chương trình trên công cụ phần mềm:

- Ngôn ngữ Java.

- Xây dựng trên thư viện đồ họa của JavaFx.

- Phần mềm IntelliJ IDEA 2019.2.3 x64.

- Phần mềm Eclipse.

1. Mục tiêu

- Hoàn thành được chương trình.

- Hoàn thành được giao diện.

- Hoàn thành được các chức năng của người dùng.

# **Chương II : Nội dung**

1. Phần mềm dùng để làm gì?

Chương trình cho phép người dùng vẽ các hình học cơ bản, xử lí số liệu từ người dùng khi nhập lệnh.

1. Dữ liệu, thông tin đầu vào

Phần mềm làm việc với những đối tượng đồ hoạ, tính toán và hiển thị chúng lên giao diện người dùng theo ý đồ và hướng dẫn đưa ra bởi user:

* Input: Thao tác chuột, lệnh dưới dạng plain text của người dùng trên giao diện làm việc của chương trình.
* Output: Hiển thị các hình học mà người dùng đã vẽ lên wordspace.

1. Mục đích, tính năng của ứng dụng

Ứng dụng bao gồm các chức năng đơn giản giúp người dùng thao tác dễ dàng, có các chức năng sau:

\*Vẽ hình:

* Đoạn thẳng (Line)
* Hình chữ nhật (Rectangle)
* Đường tròn (Circle)
* Hình bầu dục (Ellipse)
* Cung tròn (Arc)
* Đường đa tuyến (Polyline)
* Hình tam giác (Triangle)
* Hình đa giác (Polygon)

\*Thuộc tính vẽ:

* Chọn bút vẽ
* Chọn màu vẽ
* Chọn nét vẽ

\*Ghi nạp bản vẽ:  
- Lưu (Save)

- Tạo mới (New)

- Nạp (Open)

1. Giao diện dự kiến

Giao diện được tạo ra qua việc tìm hiểu, tham khảo các ứng dụng AutoCad.



Workspace

Thanh nhập lệnh

Thanh hiển thị

Thanh công cụ

*Hình 1. Giao diện chương trình*

*Bảng 1. Các Control được sử dụng trong chương trình*

|  |  |  |
| --- | --- | --- |
| TT | Tên Control | Chức năng |
| 1 | Pane | Dùng để chức các control khác như canvas, button, textfield, ….. và dùng để định hình các khu vực chức năng khác nhau trong cửa sổ làm việc của chương trình |
| 2 | Canvas | Không gian chính để làm việc,thực hiện các thao tác |
| 3 | TextField | Dùng để nhập lệnh |
| 4 | MenuBar | Bao gồm những tùy chọn New, Save, Open, Exit, About |
| 5 | ColorPicker | Dùng để chọn màu |
| 6 | Button | Dùng để thực thi các thao tác người dùng chọn loại đối tượng đồ hoạ mong muốn vẽ và các chức năng khác của chương trình |

CHƯƠNG III : PHÂN CÔNG

Sau khi tìm hiểu và quan sát nhóm đã đề ra các công việc cần làm và phân công công việc cho các thành viên như sau:

*Bảng 2. Bảng phân công công việc*

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên sinh viên | Mảng công việc | Phần trăm đóng góp |
| 1 | Bùi Thị Hồng Nhung | * Tìm hiểu đề tài * Thiết kế lớp * Thiết kế giao diện * Viết chức năng nhập lệnh * Viết báo cáo | 50% |
| 2 | Nguyễn Tấn Đạt | * Tìm hiểu đề tài * Thiết kế lớp * Viết các chức năng vẽ * Kiểm thử và debug * Viết báo cáo | 50% |

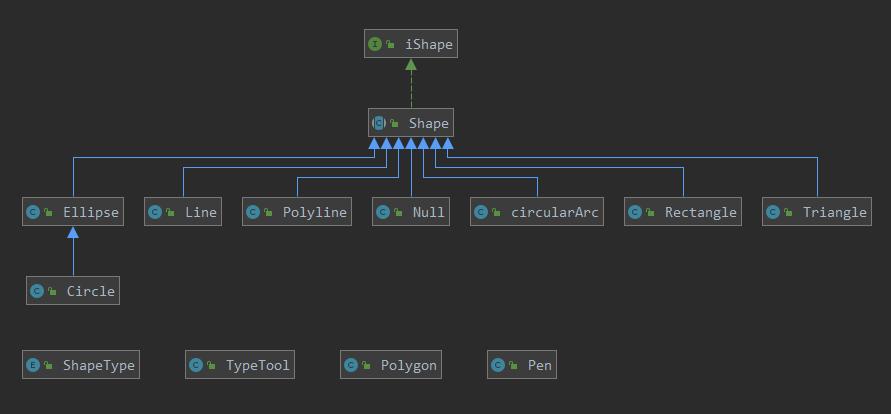
CHƯƠNG IV : THIẾT KẾ

1. ***Thiết kế lớp***

Xét thấy giữa các các đối tượng hình học có các tính chất chung, bao gồm:

* + - Điểm bắt đầu.
    - Điểm kết thúc.
    - Màu
    - Vẽ hình.

Đó là chưa kể giữa chúng còn có các phương thức và các đối tượng giống nhau. Vì thế, chúng em quyết định thiết kế class Shape, class này là base cho các class khác, chứa tất cả những phương thức, thuộc tính chung của các class. Đồng thời, các class của các đối tượng hình học khác bổ sung các phương thức, đối tượng của riêng nó để đáp ứng chức năng, nhiệm vụ của đối tượng hình học mà class đó hiện thực.



*Hình 2. Class Diagram*

* 1. ***Danh mục các lớp***

*Bảng 3. Danh mục các lớp*

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Tên lớp | Kế thừa | Mục đích |
| 1 | iShape(Interface class) |  | Quy định các behavior và properties cần thiết cho một đối tượng hình học |
| 2 | Shape (abstract class) |  | Vận dụng tính đa hình và tính thừa kế của lập trình hướng đối tượng, làm code chương trình viết ra rõ ràng và linh hoạt hơn. Khởi tạo những giao thức chính của mỗi class hình |
| 3 | Main |  | Điểm gọi chương trình |
| 4 | MainSence |  | Lớp MainSence là nơi viết các chức năng cho các Control |
| 5 | MainSence.fxml |  | Lớp MainSence.fxml dùng để chứa các dòng lệnh Tool được sinh ra |
| 6 | MainSenceStyle |  | Lớp MainSenceStyle dùng để xét những thuộc tính size, image, color cho background |
| 7 | Ellipse | Class Shape | Lớp Circle dùng để vẽ đối tượng hình tròn được kế thừa từ lớp Ellipse |
| 8 | CircularArc | Class Shape | Lớp CircularArc để vẽ hình cung tròn. |
| 9 | Circle | Class Ellipse | Lớp Ellipse dùng để vẽ hình ellipse. |
| 10 | Line | Class Shape | Lớp Line dùng đễ vẽ đường thẳng. |
| 11 | Polygon | Class Shape | Lớp Polygon dùng để tạo hình đa giác. |
| 12 | Polyline | Class Shape | Lớp Polyline dùng để tạo chuỗi đa giác. |
| 13 | Rectangle | Class Shape | Lớp Rectangle dùng để vẽ hình chữ nhật. |
| 14 | Triangle | Class Shape | Lớp Triangle dùng để vẽ hình tam giác. |
| 15 | Pen | Class Shape | Lớp Pen dùng để vẽ như bút vẽ. |
| 16 | TypeTool |  | Lớp dùng để chú thích, in ra những dòng text |
| 17 | Null | Class Shape | Lớp dùng để tạo một đối tượng hình học rỗng |

* 1. ***Đặc tả phương thức trong các lớp***
     1. *Lớp Interface (iShape)*

*Bảng 4. Phương thức của interface iShape*

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Phương thức | Mục đích | Tên file, số thứ tự dòng khai báo |
| 1 | **public** **void** draw(javafx.scene.canvas.Canvas canvas); | Dùng để vẽ đối tượng | Dòng 19  Class iShape |
| 2 | **public** **void** setFillColor(javafx.scene.paint.Color color); | Dùng để tô màu cho đối tượng | Dòng 13  Class iShape |
| 3 | **public** **void** setColor(javafx.scene.paint.Color color); | Dùng để chọn màu cho đối tượng | Dòng 9  Class iShape |
| 4 | **public** **void** setPosition(javafx.geometry.Point2D position); | Dùng để xét điểm cho đối tượng | Dòng 5  Class iShape |

* + 1. *Lớp Abstract (Shape)*

*Bảng 5. Phương thức của lớp Abstract Shape*

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Phương thức | Mục đích | Tên file số thứ tự dòng khai báo |
| 1 | **public void** setPosition(Point2D position) {  **this**.**startPosition** = position; } | Đặt vị trí điểm đầu  Tọa độ | Shape.java (44) |
| 2 | **public void** setEndPosition(Point2D position) {  **this**.**endPosition** = position; } | Đặt vị trí điểm cuối  Tọa độ | Shape.java (49) |
| 3 | **public** Point2D getPosition() {  **return this**.**startPosition**; } | Lấy vị trí điểm đầu  Tọa độ | Shape.java (54) |
| 4 | **public** Point2D getEndPosition() {  **return this**.**endPosition**; } | Lấy vị trí điểm cuối  Tọa độ | Shape.java (59) |
| 5 | **public void** setColor(Color color) {  **this**.**color** = color; } | Đặt màu cho đối tượng | Shape.java (64) |
| 6 | **public** Color getColor() {  **return this**.**color**; } | Xét màu cho đối tượng | Shape.java (69) |
| 7 | **public void** setFillColor(Color color) {  **this**.**fillColor** = color; } | Đặt fill màu cho đối tượng | Shape.java (74) |
| 8 | **public** Color getFillColor() {  **return this**.**fillColor**; } | Xét fill màu cho đối tượng | Shape.java (79) |
| 9 | **public** Point2D calculateTopLeft(){  **double** x = Math.*min*(**this**.getPosition().getX(), **this**.getEndPosition().getX());  **double** y = Math.*min*(**this**.getPosition().getY(), **this**.getEndPosition().getY());  **return new** Point2D(x,y); } | Tính điểm đầu phía bên trái khung bao quanh hình | Shape.java (99) |
| 10 | **public void** setTopLeft(Point2D pos){  **this**.**topLeft** = pos; } | Đặt điểm đầu phía bên trái khung | Shape.java (110) |
| 11 | **public** Point2D getTopLeft(){  **this**.**topLeft** = calculateTopLeft();  **return topLeft**; } | Lấy điểm đầu phía bên trái khung | Shape.java (105) |
| 12 | **public** Shape(Point2D startPos, Point2D endPos , Color strockColor){  **this**.**color** = strockColor;  **this**.**startPosition** = startPos;  **this**.**endPosition** = endPos;  **this**.**fillColor** = Color.***TRANSPARENT***;  **this**.**topLeft** = calculateTopLeft(); } | Hàm khai khởi tạo đối tượng hình | Shape.java (35) |
| 13 | **public void** draw(Canvas canvas) {  } | Vẽ đối tượng hình trên workspace Canvas | Shape.java (84) |

#### Lớp MainSence

*Bảng 6. Phương thức của lớp MainSence*

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Phương thức | Mục đích | Tên file, số thứ tự dòng khai báo |
| 1 | **public void** bnt\_Line\_Click(ActionEvent actionEvent) {  **CurrentShape** = Shape.ShapeType.***Line***;  **ClickCount** = 0; } | Chọn vẽ đường thẳng | MainSence.java (72) |
| 2 | **public void** bnt\_Ellipse\_Click(ActionEvent actionEvent) {  **CurrentShape** = Shape.ShapeType.***Ellipse***;  **ClickCount** = 0; } | Chọn vẽ hình ellipse | MainSence.java (77) |
| 3 | **public void** bnt\_Arc\_Click(ActionEvent actionEvent) {  **CurrentShape** = Shape.ShapeType.***Arc***;  **ClickCount** = 0; } | Chọn vẽ đường cong | MainSence.java (82) |
| 4 | **public void** bnt\_Triangle(ActionEvent actionEvent) {  **CurrentShape**= Shape.ShapeType.***Triange***;  **ClickCount** = 0; } | Chọn vẽ hình tam giác | MainSence.java (87) |
| 5 | **public void** bnt\_Rectagle\_Click(ActionEvent actionEvent) {  **CurrentShape** = Shape.ShapeType.***Rectangle***;  **ClickCount** = 0; } | Chọn vẽ hình chữ nhật | MainSence.java (93) |
| 6 | **public void** bnt\_Circle\_Click (ActionEvent actionEvent) {  **CurrentShape**=Shape.ShapeType.***Circle***;  **ClickCount** = 0; } | Chọn vẽ hình tròn | MainSence.java (98) |
| 7 | **public void** bnt\_Polyline\_Click(ActionEvent actionEvent) {  **CurrentShape** = Shape.ShapeType.***Polyline***;  **ClickCount** = 0; } | Chọn vẽ hình Polyline | MainSence.java (104) |
| 8 | **public void** refresh(Object canvas) {  redraw((Canvas) canvas); } |  | MainSence.java (253) |
| 9 | **public void** redraw(Canvas canvas) {  GraphicsContext gc = canvas.getGraphicsContext2D();   gc.clearRect(0, 0, 980, 640);  **try**{  gc.drawImage(**image**, 0, 0, **MainCanvas**.getWidth(), **MainCanvas**.getHeight());  **for**(**int** i=0;i<**ShapeList**.size();i++)  {  **ShapeList**.get(i).draw(canvas);  }  }**catch**(Exception e){} } |  | MainSence.java (258) |
| 10 | **public void** addShape(Shape shape) {  **ShapeList**.add(shape);  refresh(**MainCanvas**); } |  | MainSence.java (277) |
| 11 | **public void** MenuSave\_Click(ActionEvent actionEvent) **throws** IOException {  FileChooser fileChooser = **new** FileChooser();   *//Set extension filter for text files* FileChooser.ExtensionFilter extFilter = **new** FileChooser.ExtensionFilter(**"jpeg files (\*.png)"**, **"\*.png"**);  fileChooser.getExtensionFilters().add(extFilter);   *// Show save file dialog* File file = fileChooser.showSaveDialog(Main.*stage*);   **if** (file != **null**) {  **try** {  SnapshotParameters params = **new** SnapshotParameters();  params.setFill(Color.***BLACK***);  WritableImage writableImage = **new** WritableImage((**int**) **MainCanvas**.getWidth(), (**int**) **MainCanvas**.getHeight());  WritableImage snapshot = **MainCanvas**.snapshot(params, writableImage);  ImageIO.*write*(SwingFXUtils.*fromFXImage*(snapshot, **null**), **"png"**, file);  } **catch** (IOException ex)  {   }  } } | Chọn lưu lại workspace | MainSence.java (283) |
| 12 | **public void** bnt\_CmdExec\_Click(ActionEvent actionEvent) {  **CurrentShape** = Shape.ShapeType.***Polyline***;  **if**(**txt\_Cmd**.getText().equals(**""**))  {  Alert alert = **new** Alert(Alert.AlertType.***WARNING***);  alert.setTitle(**"Cảnh báo"**);  alert.setHeaderText(**"Không có lệnh nào để thực thi"**);  alert.setContentText(**"Mời nhập lệnh!"**);  alert.showAndWait();  }  **else** {  String[] arrOfStr;  String str = **txt\_Cmd**.getText().replaceAll(**"\\s"**, **""**);  arrOfStr = str.split(**"\\(|\\)|\\,"**);  **for** (String a : arrOfStr)  System.***out***.println(a);  **int** a;  **if**(arrOfStr.**length** > 0) {  **switch** (arrOfStr[0])  {  **case "Circle"**:  **case "circle"**:  **if**(arrOfStr.**length** == 2)  {  **DrawByCmd** = **true**;  **CurrentShape** = Shape.ShapeType.***Circle***;  **DrawingShape** = **new** Circle(**new** Point2D(0, 0));  **DrawingShape**.**radius** = Integer.*valueOf*(arrOfStr[1]);  **DrawingShape**.setColor(**ColorPicker**.getValue());  **DrawingShape**.setEndPosition(**new** Point2D(Math.*sqrt*(2) \* **DrawingShape**.**radius**, Math.*sqrt*(2) \* **DrawingShape**.**radius**));  **DrawingShape**.draw(**MainCanvas**);  *// addShape(DrawingShape);* }  **break**;  }  }  } } | Lệnh vẽ hình thực thi | MainSence.java (307) |
| 13 | **public void** MenuOpen\_Click(ActionEvent actionEvent) {  FileChooser fileChooser = **new** FileChooser();   *//Set extension filter for text files* FileChooser.ExtensionFilter extFilter = **new** FileChooser.ExtensionFilter(**"jpeg files (\*.png)"**, **"\*.png"**);  fileChooser.getExtensionFilters().add(extFilter);   *// Show save file dialog* File file = fileChooser.showOpenDialog(Main.*stage*);  System.***out***.println(file.toString());   **if** (file != **null**) {  **try** {  **image** = **new** Image(file.toURI().toString());  refresh(**MainCanvas**);  }  **catch** (Exception ex)  {   }  } } | Mở, truyền file ảnh lên workspace | MainSence.java (348) |
| 14 | **public void** exitApp(ActionEvent actionEvent) {  } | Dùng để thoát khỏi chương trình | MainSence.java (272) |

#### Lớp Triangle

*Bảng 7. Phương thức của lớp Triangle*

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Phương thức | Mục đích | Tên lớp và thứ tự dòng khai báo |
| 1 | **public void** AddPoint(Canvas maincanvas, Point2D point) {  **this**.**secondPoint** = point;  Line gui = **new** Line(getPosition());  gui.setEndPosition(**secondPoint**);  gui.setColor(getColor());  gui.draw(maincanvas); } | Xét thêm điểm thứ 3 | Triangle.java (32) |

* + 1. *Lớp circularArc*

*Bảng 8. Phương thức của lớp circularArc*

|  |  |  |  |
| --- | --- | --- | --- |
| TT | Phương thức | Mục đích | Tên lớp và thứ tự dòng khai báo |
| 1 | **public void** calcArcExtent() {  Point2D temp = **new** Point2D(**endPosition**.getX(), **centerPoint**.getY());  **double** oppsite = Math.*sqrt*(Math.*pow*(temp.getX() - **endPosition**.getX(), 2) + Math.*pow*(temp.getY() - **endPosition**.getY(), 2));  **arcExtent** = 180 - Math.*toDegrees*(Math.*sin*(oppsite / (**w** / 2))) - **startAngle**; } | Góc mở của đường cong | CircularArc.java (78) |
| 2 | **public void** calcStartAngle() {  Point2D temp = **new** Point2D(**startPosition**.getX(), **centerPoint**.getY());  **double** oppsite = Math.*sqrt*(Math.*pow*(temp.getX() - **startPosition**.getX(), 2) + Math.*pow*(temp.getY() - **startPosition**.getY(), 2));  **startAngle** = Math.*toDegrees*(Math.*sin*(oppsite / (**w** / 2))); } | Xác định góc để cố định điểm bắt đầu vẽ | CircularArc.java (85) |

* + 1. *Lớp Polyline*

*Bảng 9. Phương thức của lớp Polyline*

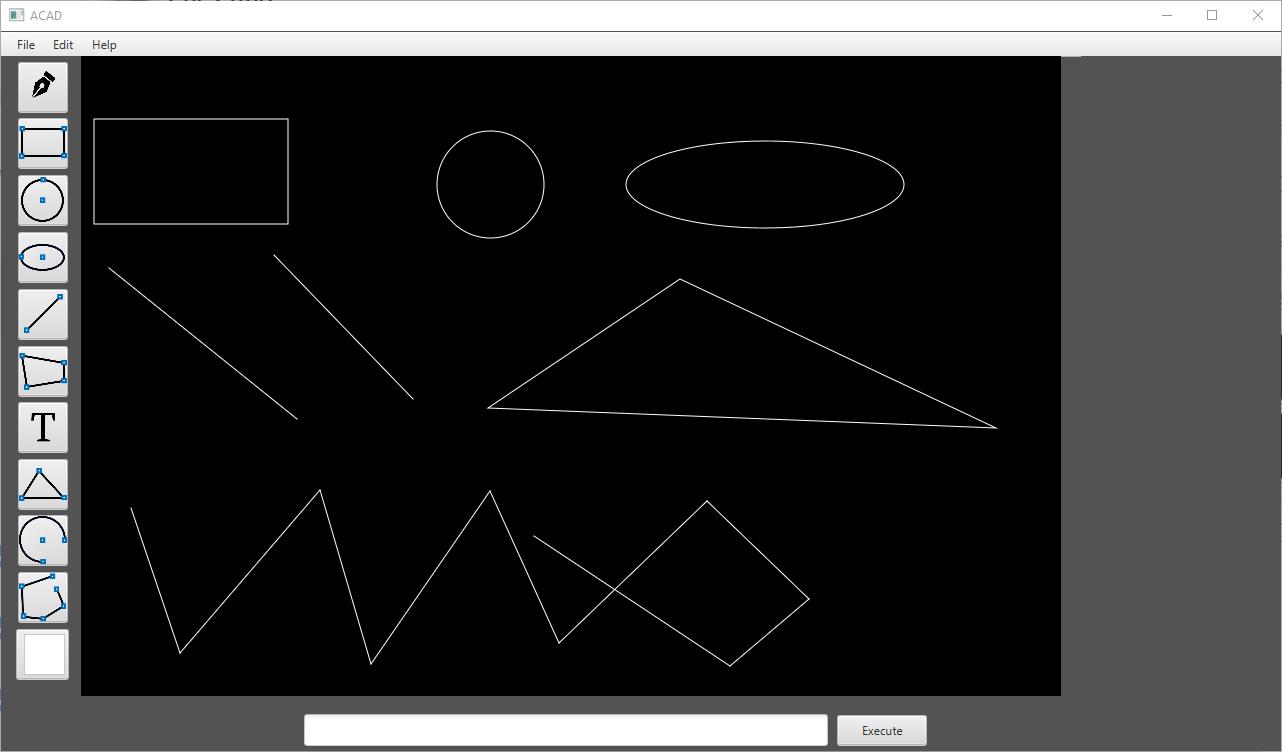
|  |  |  |  |
| --- | --- | --- | --- |
| TT | Phương thức | Mục đích | Tên lớp và thứ tự dòng khai báo |
| 1 | **public void** AddPoint(Canvas maincanvas, Point2D point) {  **Point**.add(point);  **nPoints**++;  **if**(**Point**.size() >= 2)  {  GraphicsContext gc = maincanvas.getGraphicsContext2D();  gc.setStroke(**super**.getColor());  Line gui = **new** Line(**Point**.get(**Point**.size() - 2));  gui.setEndPosition(**Point**.get(**Point**.size() - 1));  gui.setColor(getColor());  gui.draw(maincanvas);  } } | Add thêm điểm tùy theo người dùng chọn |  |

## ***Thiết kế giao diện***

*Bảng 10. Giao diện của chương trình*

|  |  |  |
| --- | --- | --- |
| Màn hình | Mục đích | Giải thích |
| *Hình 3. Giao diện chương trình* | Dùng để thực hiện các thao tác chức năng của chương trình và hiển thị hình vẽ trên wordspace. |  |

CHƯƠNG V : CÀI ĐẶT VÀ KIỂM THỬ

Các chức năng của chương trình đã được hiện thực trên workspace

*Hình 4. Giao diện hiển thị các hình được vẽ*

CHƯƠNG VI : KẾT LUẬN

## ***Kết quả đạt được***

Kết quả đạt được dựa trên các mục tiêu ban đầu đề ra, sau khi thực hiện là một chương trình vẽ được những hình học cơ bản. Hiểu rõ được các tình kế thừa trong lập trình hướng đối tượng và sử dụng được ngôn ngữ java.

*Bảng 11. Kết quả xây dựng chương trình*

|  |  |  |  |
| --- | --- | --- | --- |
| STT | Mục tiêu | Mức độ hoàn thành | Ghi chú |
| 1 | Hoàn thành được chương trình. | 60% |  |
| 2 | Hoàn thành được giao diện trang chủ. | 100% |  |
| 3 | Hoàn thành được chức năng của người dùng. | 70% |  |

1. *Kết luận*
2. *Ưu điểm*

* Chương trình đã bước đầu hoàn thành được những mục tiêu đề ra trong quá trình xây dựng ý tưởng cho chương trình.
* Đáp ứng được một số nhu cầu của người dùng, cùng với đó là chương trình có giao diện đơn giản, rõ ràng mang lại sự thuận tiện cho người sử dụng.

1. *Nhược điểm*

* Các chức năng trong chương trình còn khá đơn giản.
* Một số chức năng vẫn chưa thực thi được như: thanh hiển thị, hình vẽ Polygon, bút Pen.
* Cách thiết kế giao diện còn một số chỗ chưa được hợp lý và chuyên nghiệp.

1. *Đề xuất phương án cải thiện*

Để khắc phục được các khuyết điểm trên chương trình cần có hướng phát triển như sau:

* Xây dựng và phát triển thêm một số chức năng cho người sử dụng như: zoom hình, undo,..
* Hoàn thành những chức năng chưa hoàn thiện

PHỤ LỤC

TÀI LIỆU THAM KHẢO

Chương trình tham khảo: <https://github.com/ahmedhammad97/Javafx-Paint-Application?fbclid=IwAR2SBRul6T_Yi_xdlWDzM6n0nD7DCPIKOeU5RmhQAzVgyC26_Lx3ieztGWE>