



Windows Forms апликација изработена од:

- Стефан Кочев (151006)
- Кристијан Колев (151055)
- Јован Калаџиески (151013)

1. Опис на апликацијата

Апликацијата која ја изработивме претставува алатка за изработка на Use Case дијаграми стандардизирани со Unified Modeling Language (UML). Со имплементирање на стандардните функционалности, како и стандардниот изглед на прозорецот за ваков тип на апликации (Drawing apps), постигнуваме едноставен интерфејс за корисникот.

Како специјализирана алатка за изработка Use Case дијаграми, корисникот на DrawU (името на нашата апликација), може да исцрта исклучиво елементи кои се пропишани во UML јазикот, за моделирање на софтвер преку кориснички случаи (Use Cases). Со таквото ограничување обезбедуваме корисникот да се држи до UML стандардот, при изработката на Use Case дијаграмите, користејќи ги само стандардните елементи.

Елементите кои може да се исцртаат со оваа апликација се: **Use Case елипса** во која е испишано името на корисничкиот случај, **актер** - со кој се претставува улогата која даден ентитет ја има во софтверскиот систем и **стрелка** со која се претставуваат релациите помеѓу елементите на дијаграмот. На секоја релација се задава тип преку текст испишан покрај секоја од стрелките. Типот треба да биде еден од трите релации (пропишани во UML): **include**, **extend** и **generalization** релација. Боите на елипсите, текстот и исцртаните линии се оставени на замислата на корисникот, кој преку едноставни и интуитивни команди може да ги менува.

2. Опис на решението на проблемот

Апликацијата е имплементирана со стандарден **DV (Document View)** патерн.

Моделот на имплементацијата е претставен преку класи со кои се претставени соодветните форми кои се исцртуваат. Секоја од класите го имплементира на соодветен начин **IDrawable** интерфејсот, во кој ги дефинираме функциите. **IDrawable** интерфејсот го имплементираат класите **Line**, **UMLEllipse** и **Actor** кои ги опишуваат соодветните објекти. Сите класи се означени како **Serializable**, што овозможува моменталниот дијаграм да биде зачуван во датотека. Серијализацијата е бинарна.

- Класата **Line**:
- во следниот код е прикажан и објаснет со **xml summary** и коментари секоја променлива и функција од класата Line (прикажани се само атрибутите и потписи на функциите)

```

/// <summary>
/// Line contains all the necessary attributes for describing a line in the program.
/// It also contains methods for drawing, moving, resizing the line and helper methods
/// for achieving these functionalities.
/// </summary>

[Serializable]
public class Line : IDrawable
{
    public int X1 { get; set; }           //position along x-axis of the first point
    public int Y1 { get; set; }           //position along y-axis of the first point
    public int X2 { get; set; }           //position along x-axis of the second point
    public int Y2 { get; set; }           //position along y-axis of the second point
    public string Text { get; set; }      //Text describing the relationship between the two
elements connected with the line
    public int TextSize { get; set; }     //Size with which to write the text
    public bool Selected { get; set; }    //Indicates whether the line is selected or not
    public Color TextColor { get; set; }  //Color of text
    public Color BorderColor { get; set; } //Color of line
    public Color FillColor { get; set; }
    public int BorderWidth { get; set; }  //Thickness of line
    public int X { get { return Math.Min(X1, X2); } set { } } //position along x-axis of the leftmost point
    public int Y { get { return Math.Min(Y1, Y2); } set { } } //position along y-axis of the upmost point
    public int Height { get { return Math.Abs(Y1 - Y2); } set { } } //distance along y-axis between the two points
    public int Width { get { return Math.Abs(X1 - X2); } set { } } //distance along x-axis between the two points
    public DashStyle DashStyle { get; set; }
    public void Draw(Graphics g) {}
    public void isSelected(Point p, bool ctrl){}
    public bool IsResizeSquareClicked(Point click) {}
    public bool IsFirstPointClicked(Point click){}
    public void FirstPointResize(int dx, int dy){}

```

Класата Line соодветно ги имплемент IDrawable интерфејсот. Атрибутите се објаснети со коментари во кодот. Функцијата **Draw** претставува соодветна имплементација на Draw од IDrawable интерфејсот. **isSelected** е функција со која се проверува дали линијата е селектирана, односно дали точката аргумент припаѓа на линијата. **IsResizeSquareClicked** проверува дали е

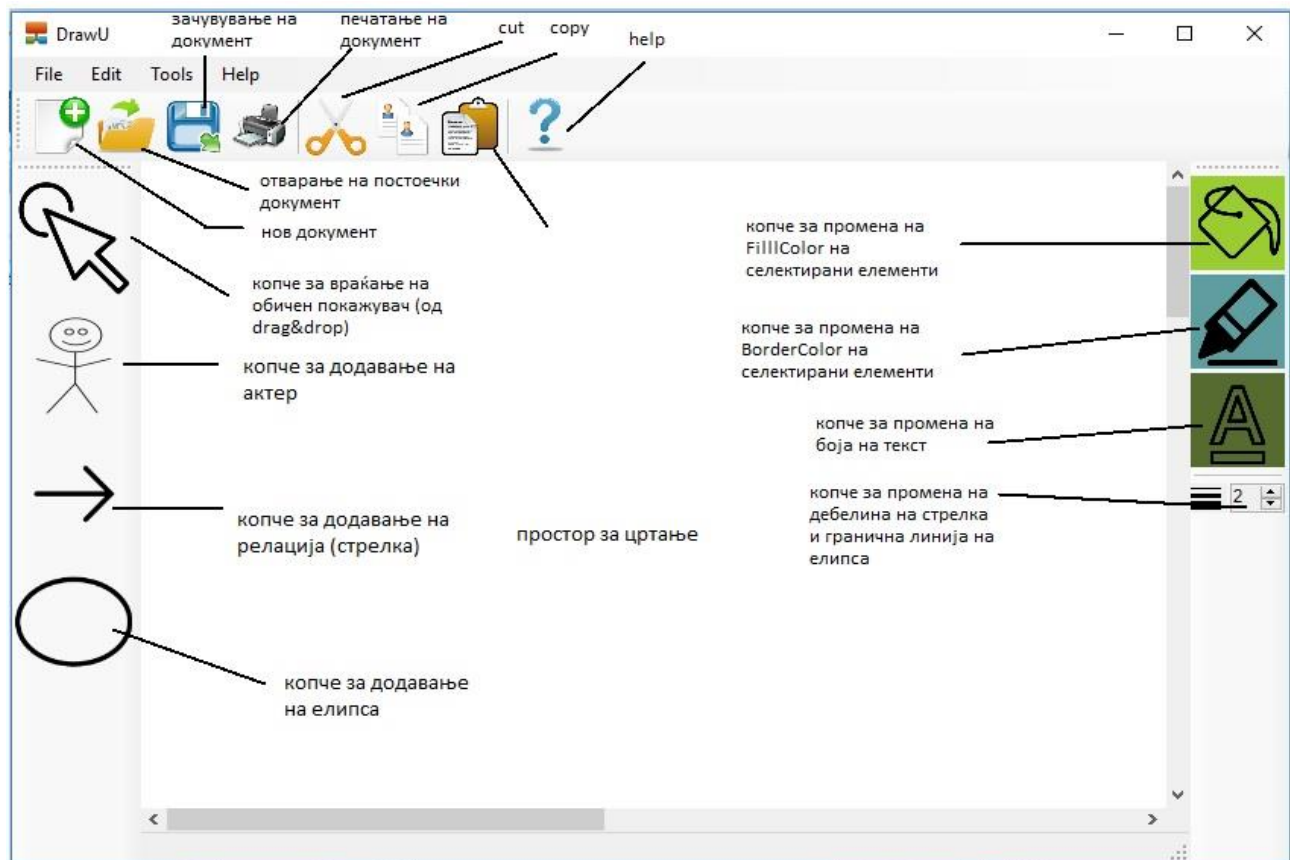
кликнат црвениот квадрат кој служи за **resize** на елементот. **IsFirstPointClicked** и **FirstPointResize** – проверка дали е клинката првата точка од линијата и промена на нејзината позиција.

Document делот од патернот е претставен преку класата **UseCaseDiagramDocument** во која во листа се чуваат сите **IDrawable** објекти, како и основните функции за манипулација со објектите во документот.

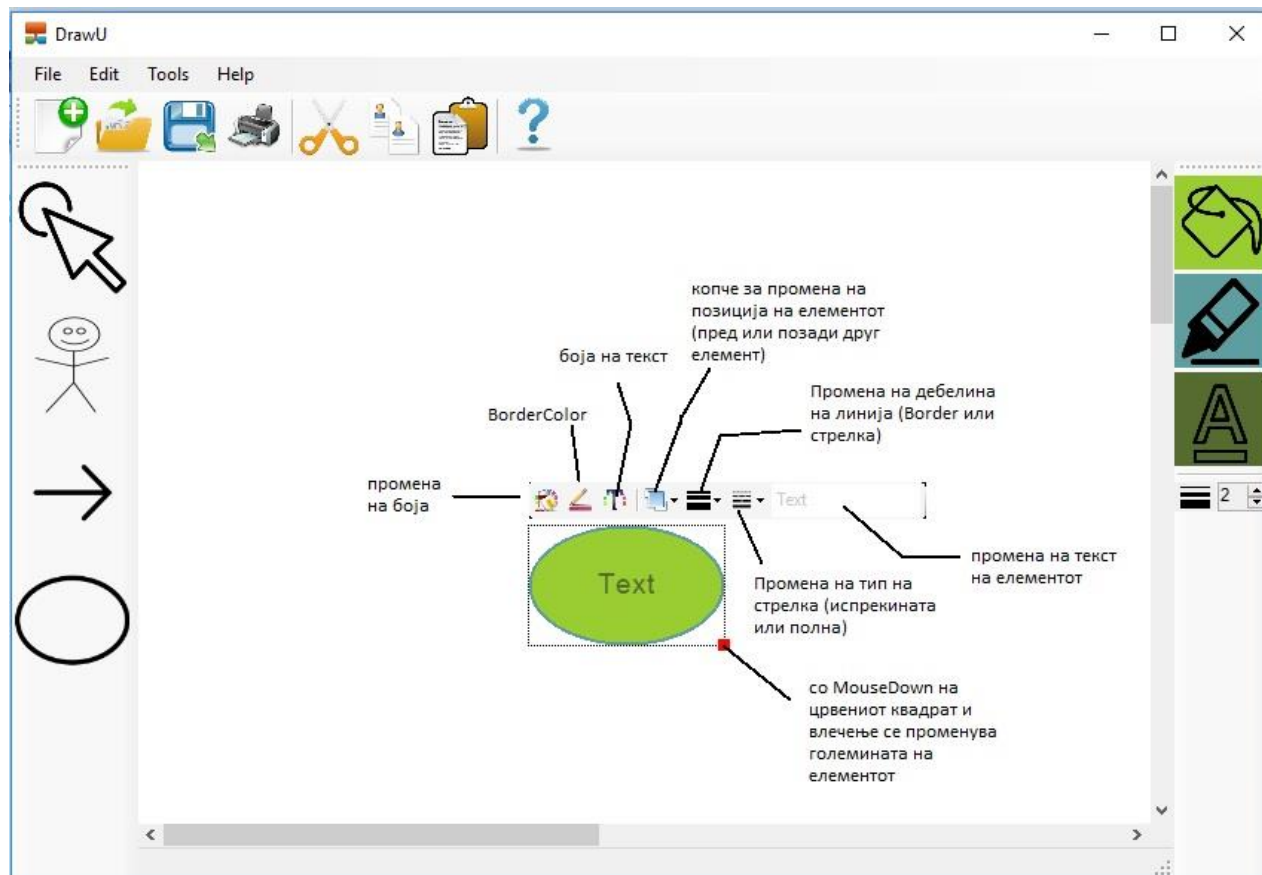
View сегментот од патернот е самиот прозорец на формата. Во класата која ја претставува формата чуваме објект од класата **UseCaseDiagramDocument** во кој е претставена целата состојба на моменталниот документ, односно дијаграм.

DocumentView патернот овозможува прегледност и модуларност на целиот код на апликацијата. Таквата архитектура на софтверот, ја прави апликацијата лесно одржлива и променлива. Промените и дополнувањата на апликацијата бараат малку промени во кодот.

3. Изглед на апликацијата и упатсво за користење



При поставување на нов елемент на површината за цртање, или при селектирање на елемент од постоечките елементи, над последниот кликнат селектиран елемент се појавува мени со основните функционалности за промена на изгледот на селектираните елементи. Менито е прикажано и објаснето на следната слика:



Кратко упатство за употреба:

- Стартување

При стартување на апликацијата се отвара главниот прозорец при што е отворен нов Blank документ.

- Вметнување на елемент

За исцртување на некој од елементите, потребно е да се кликне врз соодветниот елемент од левото вертикално мени. Со позиционирање на елементот на саканата позиција и со следен клик на десното копче од маусот се поставува елементот.

- Селектирање на елементи

Овозможено е селектирање на единечни елементи, со клик врз самите форми. Покрај селектирање на еден елемент, со задржано Ctrl копчето и кликање врз формите се овозможува селектирање на повеќе елементи. Повеќе елементи се селектираат и со селектирање со

покажувачот – се кликнува на празна површина и со задржан лев клик и движење со исцртаниот син квадрат потребно е да се опфатат формите кои сакаме да се селектираат.

- Менување на атрибути

Со копчињата од десното вертикално мени, или пак менито кое се отвара при селектирање на некоја форма, се овозможува менување на атрибутите на сите селектирани елементи. На менито кое се отвара при селектирање на форма, овозможени се промени само на атрибутите кои дадената форма ги поседува, додека останатите копчиња не може да се користат.

- Преместување на елементи

Со клик врз селектиран елемент и движење на покажувачот или со користење на копчињата лево, десно, горе или доле (стрелките на тастатурата), се овозможува промена на позицијата на сите селектирани форми.

- Зачувување, отварање на документ

За да се зачува документ потребно е да се кликне врз копчето Save од менито со алатки или пак File менито, при што се отвара Save File Dialog (стандарден Windows), кој нуди бирање на локација каде што документот ќе се зачува како и име на документот.

За отварање на постоечки документ, потребно е да се кликне на копчето Open од менито со алатки или пак File менито, при што се отвара Open File Dialog (стандарден Windows), кој нуди наоѓање на бараниот документ во датотечниот систем и негово прикажување во формата.

- Export

Со избор на Export опцијата од File менито, овозможено е експортирање на дијаграмот во jpg, png или bitmap слика.

- Cut, Copy и Paste функции

Овие стандардни функции за манипулација со формите од документот се овозможени преку контекстно мени (кое се отвара на десен клик врз работната површина), преку копчињата од менито со алатки како и со акциите Ctrl+C за Copy и Ctrl+V за Paste