У даній курсовій роботі дуже активно використовувались ООП(об’єктно-орієнтоване програмування).

Изображение выглядит как стол

Автоматически созданное описаниеОб’єктно-орієнтоване програмування – це методологія програмування засновано на представлені програми, як сукупність об’єктів, кожен із них є екземпляром певного класу, а класи утворюють ієрархію спадкування. В проєкті кожен інструмент представляється як окремий клас, при використані цих інструментів ми створюємо окремий екземпляр класу. У кожному класі прописаний свій алгоритм роботи. Так як ми реалізовуємо графічний редактор, у кожному класі задіяно бібліотеку System.Drawing. Тим самим створюємо екземпляр класу Graphics.

Для відображення створених графічних елементів створюється екземпляр класу Bitmap.

У конструкторі графічного інтерфейсу нашого програмного забезпечення ми використовуємо наступні форми: MenuStrip, ToolStripMenuItem, ToolStrip, ToolStripButton.

ToolStrip – це графічний елемент, який виконує роль панелі інструментів, в якому ToolStripButton – це інструменти створення різних елементів у графічному редакторі.

MenuStrip – це головне меню нашої програми, у головному меню присутня можливість створювати нові документи, зберігати існуючі роботи та інше.

ToolStripMenuItem – це розділи головного меню. Із додаткових елементів форми ми також використовували MainMenu, toolBar, colorDialog.

Оскільки кожен із інструментів є окремим класом, необхідно нагадати, що таке Class.

Існує три модифікатори доступу:

* private: закритий або приватний компонент класу чи структури. Приватний компонент доступний лише у межах свого класу чи структури.
* private protected: компонент класу доступний з будь-якого місця у своєму класі або похідних класах, які визначені в тій же збірці.
* protected: такий компонент класу доступний з будь-якого місця у своєму класі або у похідних класах. При цьому похідні класи можуть розміщуватися в інших зборках.
* internal: компоненти класу або структури доступні з будь-якого місця коду в тій самій збірці, однак він недоступний для інших програм та збірок.
* protected internal: поєднує функціонал двох модифікаторів protected та internal. Такий компонент класу доступний з будь-якого місця в поточній збірці та похідних класів, які можуть розташовуватися в інших зборках.
* public: публічний, загальнодоступний компонент класу чи структури. Такий компонент доступний з будь-якого місця в коді, а також інших програм і збірок.

У кожному класі існує своя власна процедура(функція void), яка виконує певний алгоритм. Окремо необхідно виділити клас ToolsClass, який містить глобальні властивості цього проєкту.