

## 3 РОЗРОБЛЕННЯ

### 3.1 ДІАГРАМА ПАКЕТІВ

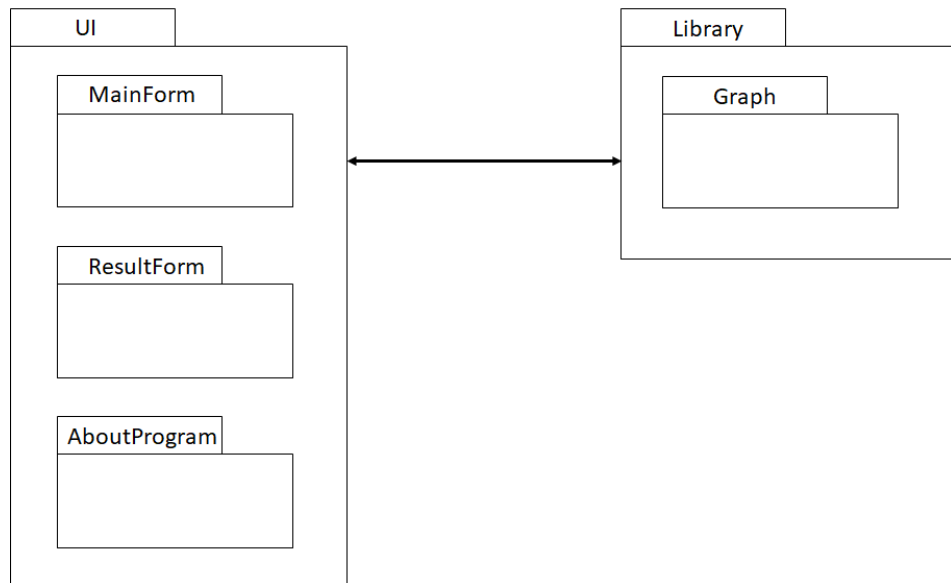


Рисунок 3.1.1 – Діаграма пакетів розробленої програми

### 3.2 ДІАГРАМА КЛАСІВ

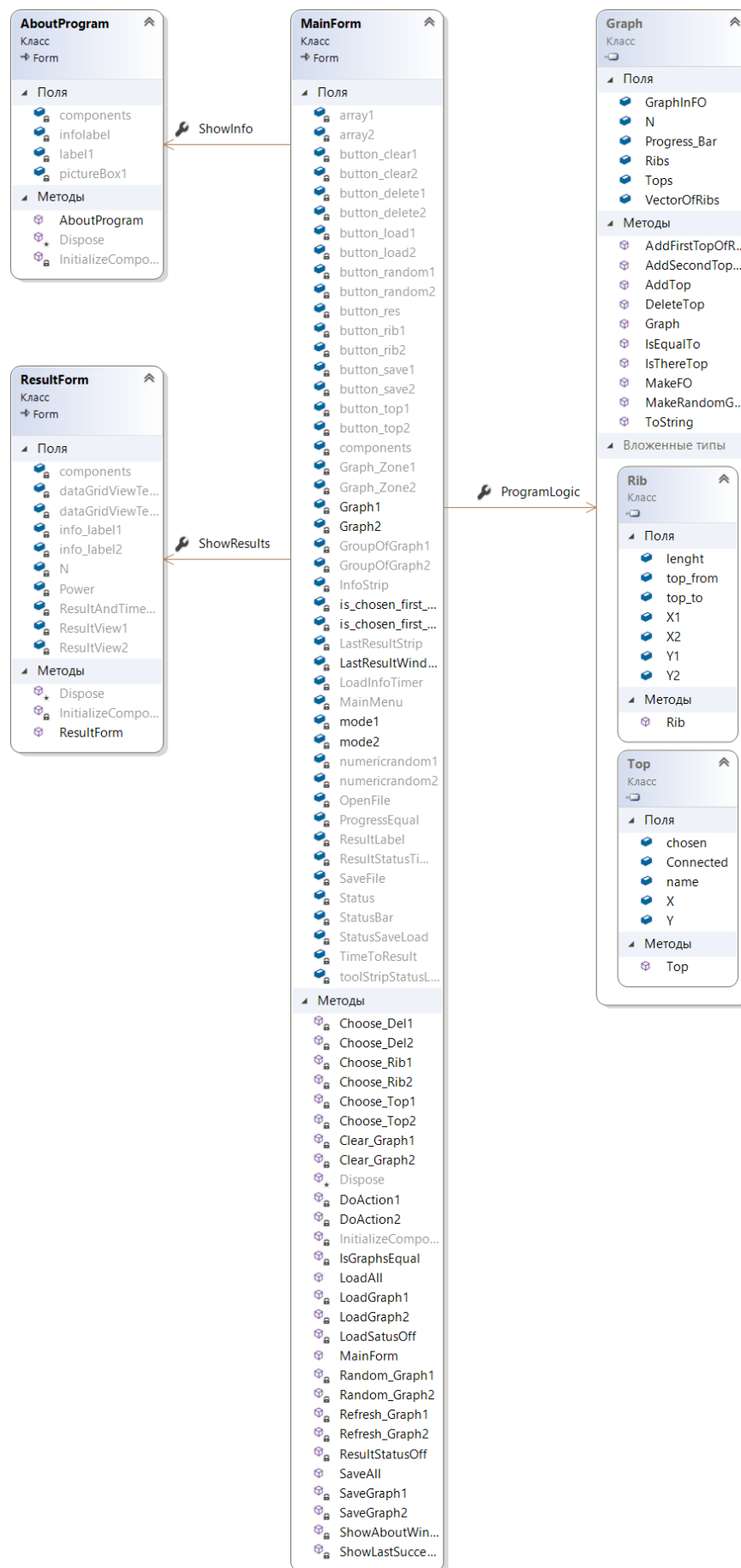


Рисунок 3.2.1 – Діаграма класів розробленої програми

### 3.3 КЛЮЧОВІ МЕТОДИ КЛАССІВ

**\*** – метод повторюється, замість цього символу може стояти цифра 1 або 2, в залежності від того, для якого графа буду застосовуватися цей метод, для першого або для другого.

№	Назва	Призначення	Сигнатура	Призначення вхідних і вихідних параметрів
1	MakeFO	Трансформує поточний граф в представлення FO	<code>public void MakeFO()</code>	–
2	AddTop	Додає нову вершину у граф	<code>public void AddTop(int X, int Y)</code>	X Y – фізичне положення вершини (місце кліку користувача)
3	AddFirstTopOfRib	Додає у буфер першу вершину ребра графу	<code>public void AddFirstTopOfRib(int name)</code>	name – назва першої вершини у ребрі
4	AddSecondTopOfRib	Додає нове ребро у граф на основі першої вершини у буфері	<code>public void AddSecondTopOfRib(int name)</code>	name – назва другої вершини у ребрі
5	DeleteTop	Видаляє задану вершину та всі прилеглі до неї ребра у графі	<code>public void DeleteTop(int name)</code>	name – назва вершини для видалення
6	IsEqualTo	Перевіряє графи на еквівалентність	<code>public bool IsEqualTo(Graph second_graph)</code>	second_graph – граф для порівняння, повертає true, якщо графи еквівалентні
7	MakeRandomGraph	Генерую випадковий граф	<code>public void MakeRandomGraph(byte highs)</code>	highs – кількість вершин
8	IsThereTop	Перевіряє, чи є в тому місці, де натиснув користувач вершина	<code>public int IsThereTop(int X, int Y)</code>	X Y – місце кліку користувача, повертає назву вершини, якщо вона є, повертає -1, якщо в цьому місці немає вершин
9	ToString	Повертає FO представлення поточного графу	<code>public override string ToString()</code>	–
10	IndexOfTop	Повертає індекс вершини по її назві	<code>private int IndexOfTop(int name)</code>	name – назва вершини, повертає індекс вершини name
11	IsExist	Перевіряє дублікати нових ребер	<code>private bool IsExist(Rib newrib)</code>	newrib – нове ребро у графі, повертає false якщо такого ребра немає та true якщо є
12	SearchLastTop	Рекурсивний метод для пошуку повних підграфів	<code>private void SearchLastTop(ref uint[] VectorOfRibs, ref byte count_of_cycles, ref List&lt;byte&gt; Used_Tops, byte pre_top, ref List&lt;int&gt;[] Tops_In_Fo, ref uint count, uint power)</code>	VectorOfRibs – посилання на вектор повних підграфів, count_of_cycles – потужність підграфу, Used_Tops – колекція використаних вершин,

				pre_top – попередня знайдена вершина, Tops_In_FO – колекція графу у представленні FO, count – кількість рекурсивно визваних методів, max_power – поточно шукана потужність графу
13	CopyCollection	Копіює колекцію в задану	<code>private List&lt;byte&gt; CopyCollection(List&lt;byte&gt; Collection, uint length)</code>	Повертає колекцію з length перших елементів колекції Collection
14	IsMatrixesAreEqual	Порівнює два вектори на еквівалентність	<code>private bool IsMatrixesAreEqual(uint[] Matrix1, uint[] Matrix2)</code>	Повертає true, якщо вектори Matrix1 та Matrix2 рівні, та false якщо ні

Таблиця 3.3.1 – Методи класу “Graph”

№	Назва	Призначення	Сигнатура	Призначення вхідних і вихідних параметрів
1	IsGraphsEqual	Перевірка графів на еквівалентність	<code>private void IsGraphsEqual(object sender, EventArgs e)</code>	Кнопка для порівняння графів
2	LoadAll	Десеріалізація графу з файлу *.JSON	<code>public Graph LoadAll(string path)</code>	Повертає десеріалізований об’єкт класу Graph з файлу, який знаходиться за адресою path
3	SaveAll	Серіалізація у файл *.JSON	<code>public void SaveAll(Graph graph, string path)</code>	Серіалізує екземпляр класу Graph за адресою path
4	LoadSatusOff	Сховати повідомлення про успішне завантаження графу с пам’яті	<code>private void LoadSatusOff(object sender, EventArgs e)</code>	—
5	ResultStatusOff	Сховати повідомлення про успішне порівняння графів	<code>private void ResultStatusOff(object sender, EventArgs e)</code>	—
6	ShowAboutWindow	Показати вікно «Інфо»	<code>private void ShowAboutWindow(object sender, EventArgs e)</code>	—
7	ShowLastSuccessfulResult	Показати вікно подробиць результату	<code>private void ShowLastSuccessfulResult(object sender, EventArgs e)</code>	—
8	Choose_Top*	Вибрати режим «встановити вершину»	<code>private void Choose_Top*(object sender, EventArgs e)</code>	—
9	Choose_Rib*	Вибрати режим «встановити ребро»	<code>private void Choose_Rib*(object sender, EventArgs e)</code>	—

<b>10</b>	Choose_Del*	Вибрати режим «видалити вершину»	<code>private void Choose_Del*(object sender, EventArgs e)</code>	—
<b>11</b>	Clear_Graph*	Видалити весь граф	<code>private void Clear_Graph*(object sender, EventArgs e)</code>	—
<b>12</b>	SaveGraph*	Зберегти граф	<code>private void SaveGraph*(object sender, EventArgs e)</code>	—
<b>13</b>	LoadGraph*	Завантажити граф з пам'яті	<code>private void LoadGraph*(object sender, EventArgs e)</code>	—
<b>14</b>	Refresh_Graph*	Подія «Paint» для графічного відображення графа на екрані	<code>private void Refresh_Graph*(object sender, PaintEventArgs e)</code>	—
<b>15</b>	DoAction*	Виконати дію відносно вибраного режиму	<code>private void DoAction*(object sender, MouseEventArgs e)</code>	—
<b>16</b>	Random_Graph*	Генерація випадкового графу	<code>private void Random_Graph*(object sender, EventArgs e)</code>	—

Таблиця 3.3.2 – Методи класу “MainForm”