|  |  |  |
| --- | --- | --- |
| Министерство образования Республики Беларусь  Учреждение образования  Белорусский Государственный Университет Информатики и Радиоэлектроники | | |
| Факультет  Кафедра | Компьютерных сетей и систем  Информатики | |
|  |  | |
|  |  | |
| **ЛАБОРАТОРНАЯ РАБОТА**  по курсу «Современные технологии разработки программного обеспечения»  **КЛАСТЕРИЗАЦИЯ ДАННЫХ ПО НЕЧИСЛОВЫМ ПРИЗНАКАМ** | | |
| Студент:  гр. 7M2632  Тузик А.С. |  | Проверил:  Стержанов М.В. |
| Минск, 2018 | | |

**ПОСТАНОВКА ЗАДАЧИ**

Целью данной лабораторной работы является разработка системы, предназначенной для кластеризации данных по нечисловым (строковым) признакам и визуализации этих данных в соответствии с результатами кластеризации.

Для достижения цели лабораторной работы были поставлены следующие задачи:

1. Реализация получения и предварительной обработки данных.
2. Реализация алгоритма кластеризации.
3. Реализация модуля визуализации полученных результатов.

**ХОД РАБОТЫ**

В качестве инструментов реализации лабораторной работы была выбрана технология UWP, C#. Была выбрана трёхслойная структура приложения, каждый из слоёв реализовавлся в виде отдельного проекта:

**UI:** STRPO.UI (UWP)

**Logic:** STRPO.Clusterization

(Clustering Algorithm)

**Data Access:** STRPO.Data

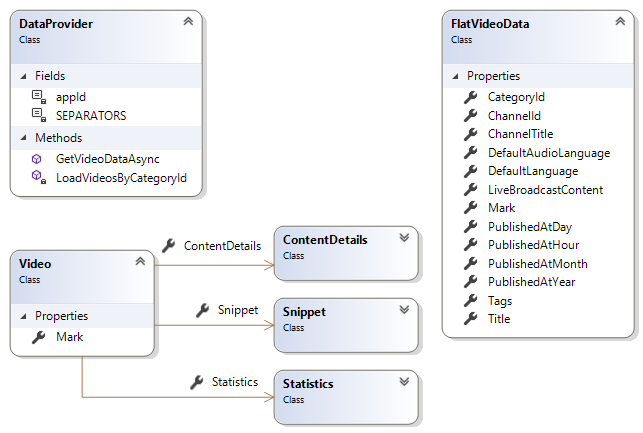
**YouTube Data API v3**

Структура приложения

***Получение и предварительная обработка данных***

В качестве данных для кластеризации я выбрал данные о видео, доступные используя YouTube Data API v3. Данные загружаются в формате JSON, из них формируются объекты, содержащие поля, которые будут в дальнейшем использоваться в качестве признаков для кластеризации.

Как результат реализации получения и предварительной обработки данных был получен модуль STRPO.Data, который пожно представить следующей диаграммой классов:



На диаграмме изображены реализованные классы:

* DataProvider – используется для получения данных о видео и преобразования их из формата JSON в модели
* Video, Snippet, Statistics, ContentDetails – соответствующие получаемым с сервера объектам модели данных
* FlatVideoData – модель данных, используемая в дальнейшем для доступа к свойствам кластеризуемых объектов используя методы System.Reflection.

***Алгоритм кластеризации***

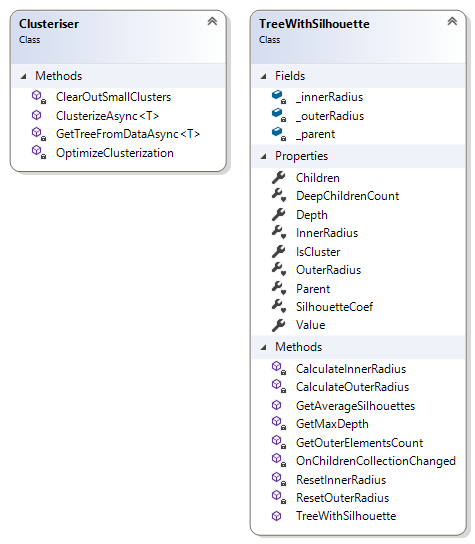
Для того, чтобы выполнить кластеризацию по нечисловым признакам я построил расстояние между объектами кластеризируемых данных, объединив их в дерево.

Для этого я упорядочил признаки по убыванию важности. На каждом шаге построения дерева я брал очередной признак, получал все известные его значения, таким образом получая количество дочерних вершин дерева. При этом, если объект не обладает признаком, считаем, что значение этого признака на объекте является значением по умолчанию.

Таким образом каждому листу дерева соответствюет один или несколько кластеризируетмых объектов.

Чтобы получить кластеры вычисляем значения коэффициента силуэта для вершины дерева и её дочерних вершин, разбивая их на отдельные кластеры, в случае если среднее арифметическое значений коэффициента силуэта дочерних вершин больше, чем у родительской.

Как результат реализации был получен модуль STRPO.Clusterisation, который пожно представить следующей диаграммой классов:



На диаграмме изображены реализованные классы:

* Clusteriser – используется для получения из коллекции элементов объект типа TreeWithSilhouette, используя алгоритм кластеризации, методы System.Reflection и многопоточную рекурсию
* TreeWithSilhouette – модельданных, реализующая структуру данных «дерево» и предоставляющая дополнительную информацию ней.

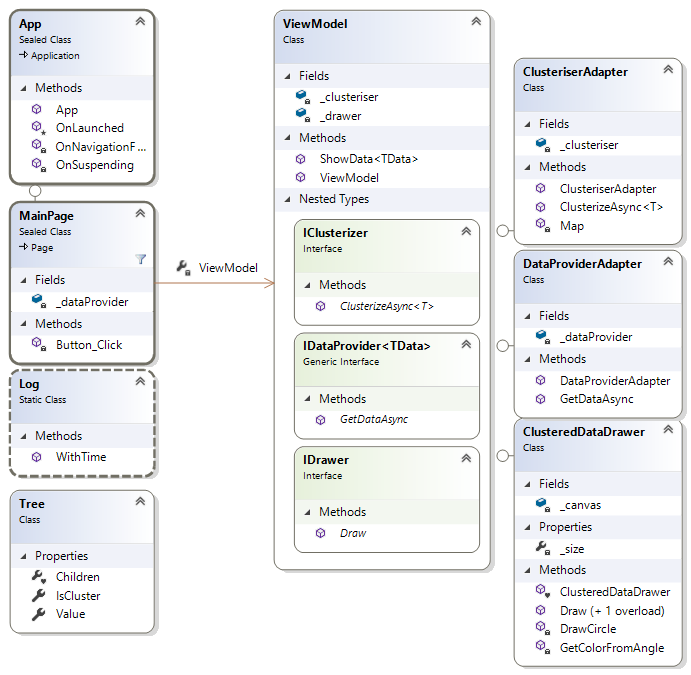
***Визуализация***

В качестве средства визуализации я решил написать приложение на C#, спользуя технологию .NET UWP.

Листья, полученного в результате кластеризации дерева, которые соответствуют кластеризируемым объектам отображаются в виде цветных точек на экране, с возможностью отображения дополнительной информации по нажатию правой кнопкой мыши на соответствующую точку.

Вершины отображаемого дерева рисуются на виртуальных окружностях, что позволяет отобразить построенное на этапе кластеризации расстояние.

Как результат реализации был получен модуль STRPO.UI, который пожно представить следующей диаграммой классов:



На диаграмме изображены реализованные классы:

* App, MainPage – стандартные классы приложения UWP для отображения пользовательского интерфейса
* Log – вспомогательный класс для отладки приложения
* ViewModel – класс инкапсулирующий в себя логику взаимодействия между собой компонент приложения, также объявляет интерфейсы зависимостей, требуемые для работы приложения
* ClusteredDaraDrawer – уласс отвечающий за отрисовку кластеризованных данных
* ClusteriserAdapter, DataProviderAdapter – классы, приводящие классы из других модулей: Clusteriser и DataProvider, соответственно, к интерфейсам зависимостей класса ViewModel

**ЗАКЛЮЧЕНИЕ**

В процессе выполнения лабораторной работы мною получены практические навыки по работе с публичным API с целью получения данных, а также их обработке для дальнейшей кластеризации, навыки применения методов System.Reflection. Применены паттерны проектирования (интерфейс, адаптер, внедрение зависимостей). Приобретён дополнительный опыт работы с технологией UWP и использованием её для отображения графики.

Разаботанные программные модули кластеирзации и визуализации данных в дальнейшем могут быть использованы для обработки других данных. Планируется расширить использемых алгоритм для кластеризации одновременно по числовым и нечисловым признакам.