Щербаков Артём Юрьевич, БПИ-244-1, вариант **5-BS**

**Общая структура меню:**

1. "Просмотреть список городов",
2. "Информация о городе",
3. "Города на карте",
4. "Добавить город",
5. "Редактировать город",
6. "Удалить город",
7. "Просмотреть пропущенные строк", *(в случае наличия)*
8. "Ввести новые данные",
9. "Сохранить",
10. "Выйти"
11. Просмотр введенных данных о городах в табличном виде
12. Просмотр информации об отдельном городе с подгрузкой географических спецификаций и погоды в месте, заданном координатами
13. Вывод городов в виде карты ASCII и точек-маркеров на ней
14. Добавление города в данные программы. Реализовано предложение координат в случае совпадения с географическим объектом
15. Редактирование города. Реализовано автозаполнение полей ввода прошлыми данными при нажатии “Enter” *(предложение выведено в скобках)*
16. Удаление города из данных программы **без подтверждения**
17. Просмотр пропущенных строк при парсинге. Появляется только при наличии необработанных строк файла (необработанной строкой считается строка, в которой в данных заданного формата допущена ошибка: "latitude": 35.6w895,)
18. Ввод данных из другого файла **без сохранения предыдущих данных**
19. Сохранение данных программы во входной файл без выхода из программы
20. Выход из программы без сохранения

**Инструкция по работе с приложением:**

1. Запустите приложение
2. Введите путь к существующему файлу (в случае несуществующего файла программа оповестит вас о недостойности вашего поступка)
3. Выберите формат разметки файла входных данных. В случае ошибки парсинга **CSV** программа выскажет вам свое недовольство, скушает все, что сможет, и предложит просмотреть строки, не попавшие в ее данные. В случае ошибки парсинга **JSON** программа не примет файл целиком, но только если в каком-то объекте допущена ошибка типа. Оба варианта парсинга пропустят несуществующие поля и выставят пустые строки и N/A для чисел в нужные данные города. Стоит отметить, что отсутствие координат – повод для пропуска всей строки или ошибки сериализации для JSON, так как они являются обязательным параметром.
4. **Работа с интерфейсом:**
5. Поле автозаполнения: строка вида: «Введите новое название города: (Нью-Йорк):» реализует автозаполнение значением из скобок при нажатии на Enter
6. Множественный выбор: переключение между пунктами и данными в формате списка осуществляется с помощью стрелок «Вверх» и «Вниз» клавиатуры. Выбор пункта – Enter
7. Для сохранения, обновления данных, выхода выберите соответствующие пункты меню. Программа сохраняет обработанные и измененные данные в файл-источник. В случае, если коварный проверяющий переместил файл за время работы программы, будет создан новый файл с соответствующим изначальному именем.

**Формат входных данных**

Программа принимает на вход JSON- и CSV-файлы, выбор разметки производится пользователем при вводе файла

Ожидаемая структура JSON (без номеров строк):

1. [
2. {
3. "name": "\u0422\u043E\u043A\u0438\u043E",
4. "country": "\u042F\u043F\u043E\u043D\u0438\u044F",
5. "population": 13929286,
6. "latitude": 35.6895,
7. "longitude": 139.6917
8. },

Для избежания конфликтов кодировки запись кириллицы происходит в виде Unicode-символов. На входе программа одинаково корректно считывает и последовательность Unicode, и нормальные названия.

Поле «маркер» в JSON не попадает.

Ожидаемая структура CSV (без номеров строк):

1. Name,Country,Population,Latitude,Longitude,Marker
2. Токио,Япония,13929286,35.6895,139.6917,1
3. Москва,Россия,12655050,55.7558,37.6176,2

Маркер – служебное необязательное поле, которое нужно только для отрисовки городов на карте. По умолчанию заполняется автоматически и сохраняется в файл на всякий случай для удобства работы с картой (например, если вы ее сохранили).