

Задание №1

Разработка программного приложения

«Редактор иерархической структуры данных, описанной в формате json»

1 Общие сведения

Требуется разработать графическое приложение, предназначенное для редактирования наборов данных, описываемых в виде иерархической структуры в формате json.

Иерархическая структура состоит из элементов трех типов – *station*, *arm* и *device*. Корневым элементом иерархии является массив элементов типа *station*. Поле *items* элемента типа *station* содержит в себе массив (возможно пустой) элементов типа *arm*. Поле *items* элемента типа *arm* содержит в себе массив (возможно пустой) элементов типа *device*.

Набор полей элементов иерархической структуры представлен в таблице 1.

Таблица 1 – Набор полей элементов иерархической структуры

Поле	Тип	Значение
Корневой элемент		
-	Массив	Массив элементов типа station
Элемент типа station		
-	-	-
type	string	Station
id	string	Строка-идентификатор, уникальная в пределах файла
name	string	Человеко-читаемое название
items	Массив	Массив элементов типа arm
Элемент типа arm		
type	string	arm
id	string	Строка-идентификатор, уникальная в пределах файла
name	string	Человеко-читаемое название
items	Массив	Массив элементов типа device
Элемент типа device		
type	string	device
id	string	Строка-идентификатор, уникальная в пределах файла
name	string	Человеко-читаемое название

Пример json-файла представлен в приложении А.

Приложение должно обеспечивать:

1.1 Загрузку иерархической структуры из json-файла заданного формата и ее отображения в виде дерева. Пример отображения представлен в приложении Б. Желательна возможность сворачивания / разворачивания элементов дерева.

1.2 Возможность редактирования иерархической структуры и ее полей в графическом интерфейсе:

- удаление элемента;
- добавление элемента;
- копирование элемента;
- редактирование полей элемента.

1.3 Сохранение результатов редактирования в старый или вновь созданный json-файл.

2 Прочие требования

2.1 Размер входного файла – до 10 000 элементов иерархической структуры.

2.2 Требования к технологии / языку программирования не предъявляются.

2.3 При разработке желательно использование системы контроля версий.

Приложение А

Пример json-файла

```
[
  {
    "name": "Станция-1",
    "type": "station",
    "id": "station-1",
    "items": [
      {
        "name": "Станция-1 Арм-1",
        "type": "arm",
        "id": "arm-1-1",
        "items": [
          {
            "name": "Станция-1 Арм-1 Устройство-1",
            "type": "device",
            "id": "d-1-1-1"
          },
          {
            "name": "Станция-1 Арм-1 Устройство-2",
            "type": "device",
            "id": "d-1-1-2"
          }
        ]
      }
    ]
  },
  {
    "name": "Арм-2",
    "type": "arm",
    "id": "arm-1-2",
    "items": [
      {
        "name": "Арм-2 Устройство-1",
        "type": "device",
        "id": "d-1-2-1",
        "items": []
      },
      {
        "name": "Арм-2 Устройство-2",
        "type": "device",
        "id": "d-1-2-2"
      }
    ]
  }
],
{
  "name": "Станция-2",
  "type": "station",
  "id": "station-2",
  "items": [
    {
      "name": "Станция-2 Арм-1",
      "type": "arm",
      "id": "arm-2-1",
      "items": [
        {
          "name": "Станция-2 Арм-1 Устройство-1",
          "type": "device",
          "id": "d-2-1-1"
        }
      ]
    },
    {
```

```
        "name": "Станция-2 Арм-1 Устройство-2",
        "type": "device",
        "id": "d-2-1-2"
    }
]
}
```

Приложение Б

Пример отображения иерархической структуры

- ▲ Станция-1
 - ▲ Станция-1 Арм-1
 - Станция-1 Арм-1 Устройство-1
 - Станция-1 Арм-1 Устройство-2
 - ▲ Арм-2
 - Арм-2 Устройство-1
 - Арм-2 Устройство-2
- ▲ Станция-2
 - ▲ Станция-2 Арм-1
 - Станция-2 Арм-1 Устройство-1
 - Станция-2 Арм-1 Устройство-2

Unit Арм-2 Устройство-1	
<div><div><div></div><div>z</div><div>↓</div></div><div>Search</div><div></div></div>	
Id	d-1-2-1
Name	Арм-2 Устройство-1

Рисунок Б.1 – Пример отображения иерархической структуры