# Онтология для сетей ЭВМ

Хлопков Дмитрий P41171

# Зачем?

1. Множество сетевых протоколов на разных уровнях модели OSI, которые могут существовать в рамках одной сети

2. Протоколы разных уровней имеют свои характеристики: частота, максимальная дальность, пропускная способность, итд.

3. Для работы протоколов устройству может быть необходимо дополнительное оборудование (сетевая карта Ethernet, Wi-Fi модуль)

4. Возможность создания унифицированного формата (например для дальнейшей визуализации)

# Для кого?

1 IOT-технологии

2 Облачные технологии

3 Архитекторы сетей

# Источники данных

1 Организации занимающиеся стандартизацией сетевых протоколов:

- W3C
- IEEE
- ISO
- IETF

2 Существующие базы знаний

- DBpedia
- Wikidata

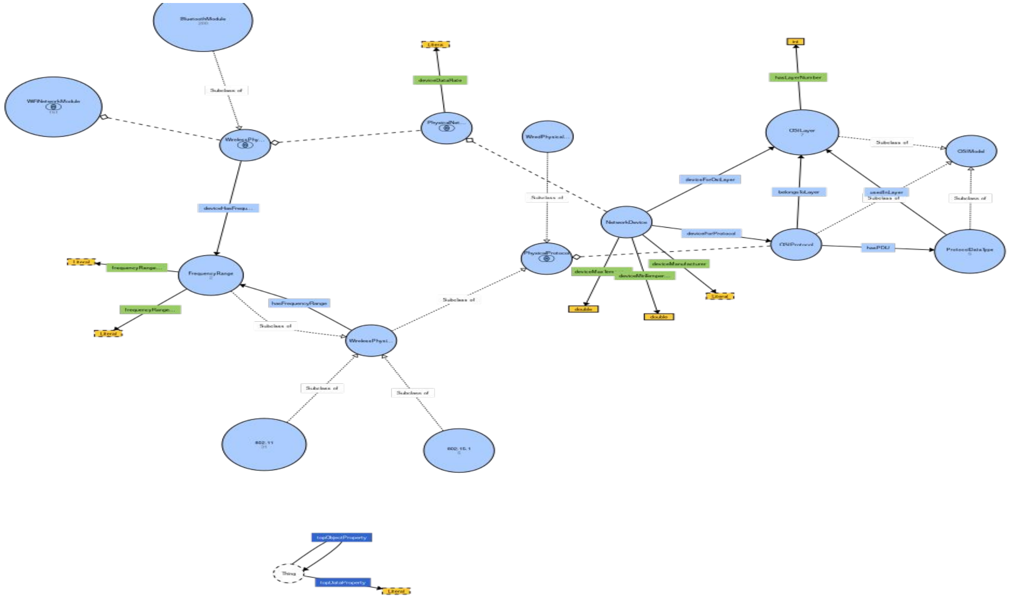
3 Магазины комплектующих

- mouser.com
- amperka.ru

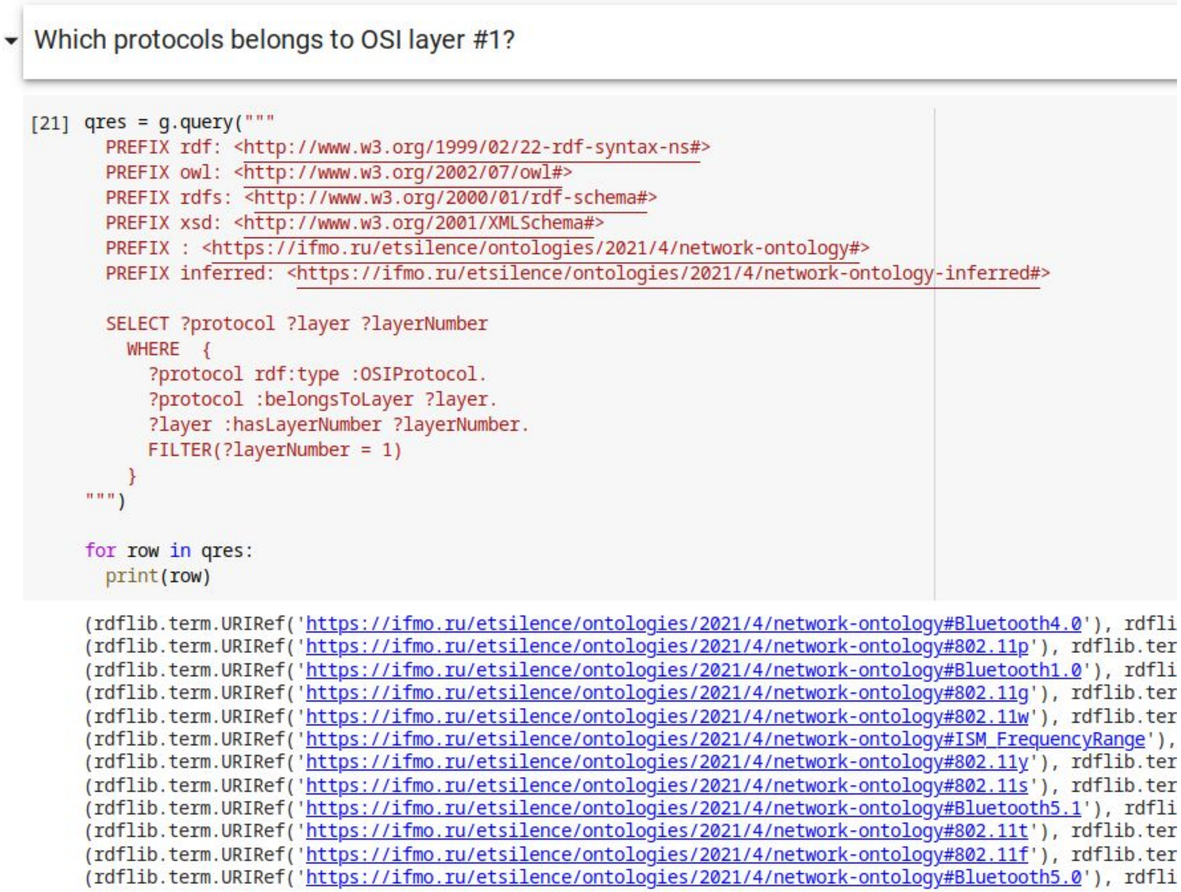
# Компетентностные вопросы

- Какие протоколы существуют на уровне OSI модели X?
- Какие устройства могут понадобиться для обеспечения протокола X?
- Какие устройства могут работать при температуре X?
- Какие устройства могут работать на частоте X?
- Какие производители выпускают устройства для протоколов X?

# Разработанная онтология



# SPARQL



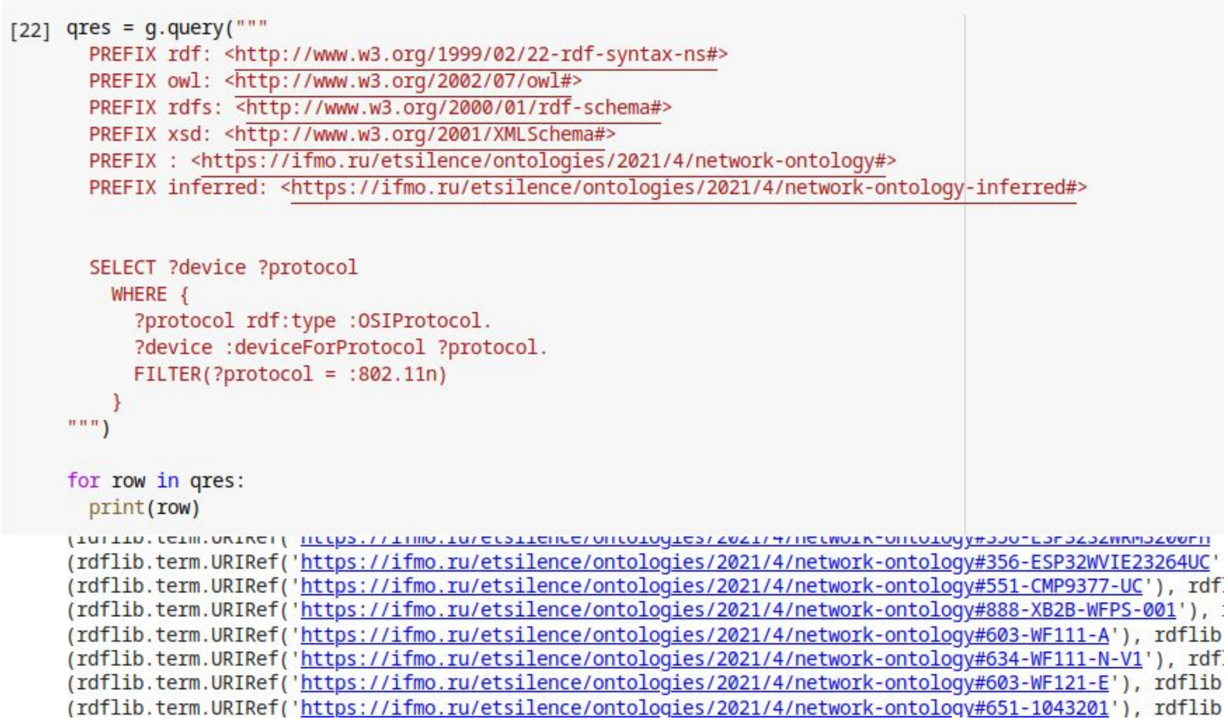
Which protocols belongs to OSI layer #1?

```
[21] qres = g.query("""
       PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
       PREFIX owl: <http://www.w3.org/2002/07/owl#>
       PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
       PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
       PREFIX : <https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#>
       PREFIX inferred: <https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology-inferred#>

       SELECT ?protocol ?layer ?layerNumber
         WHERE  {
           ?protocol rdf:type :OSIProtocol.
           ?protocol :belongsToLayer ?layer.
           ?layer :hasLayerNumber ?layerNumber.
           FILTER(?layerNumber = 1)
         }
     """)

     for row in qres:
       print(row)
```

```
     (rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#Bluetooth4.0'), rdfli
     (rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#802.11p'), rdflib.ter
     (rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#Bluetooth1.0'), rdfli
     (rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#802.11g'), rdflib.ter
     (rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#802.11w'), rdflib.ter
     (rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#ISM_FrequencyRange'),
     (rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#802.11y'), rdflib.ter
     (rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#802.11s'), rdflib.ter
     (rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#Bluetooth5.1'), rdfli
     (rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#802.11t'), rdflib.ter
     (rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#802.11f'), rdflib.ter
     (rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#Bluetooth5.0'), rdfli
```

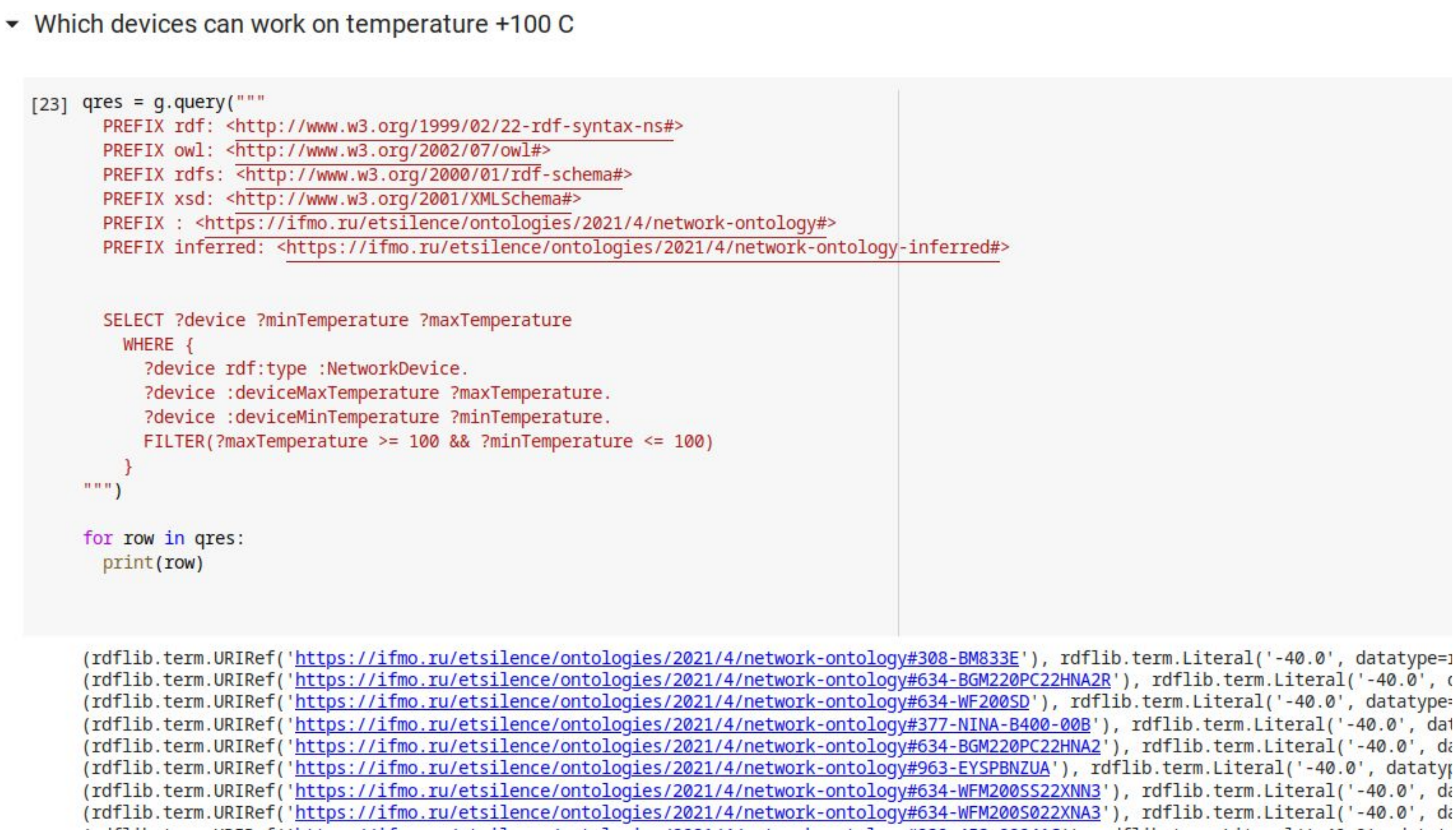
# SPARQL

▾ Which devices support 802.11n protocol?



```
[22] qres = g.query("""
       PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
       PREFIX owl: <http://www.w3.org/2002/07/owl#>
       PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
       PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
       PREFIX : <https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#>
       PREFIX inferred: <https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology-inferred#>


       SELECT ?device ?protocol
         WHERE {
           ?protocol rdf:type :OSIProtocol.
           ?device :deviceForProtocol ?protocol.
           FILTER(?protocol = :802.11n)
         }
     """)

     for row in qres:
       print(row)
     (rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#356-ESP32WRM3200FH
     (rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#356-ESP32WVIE23264UC'
     (rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#551-CMP9377-UC'), rdf
     (rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#888-XB2B-WFPS-001'),
     (rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#603-WF111-A'), rdflib
     (rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#634-WF111-N-V1'), rdf
     (rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#603-WF121-E'), rdflib
     (rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#651-1043201'), rdflib
```

# SPARQL

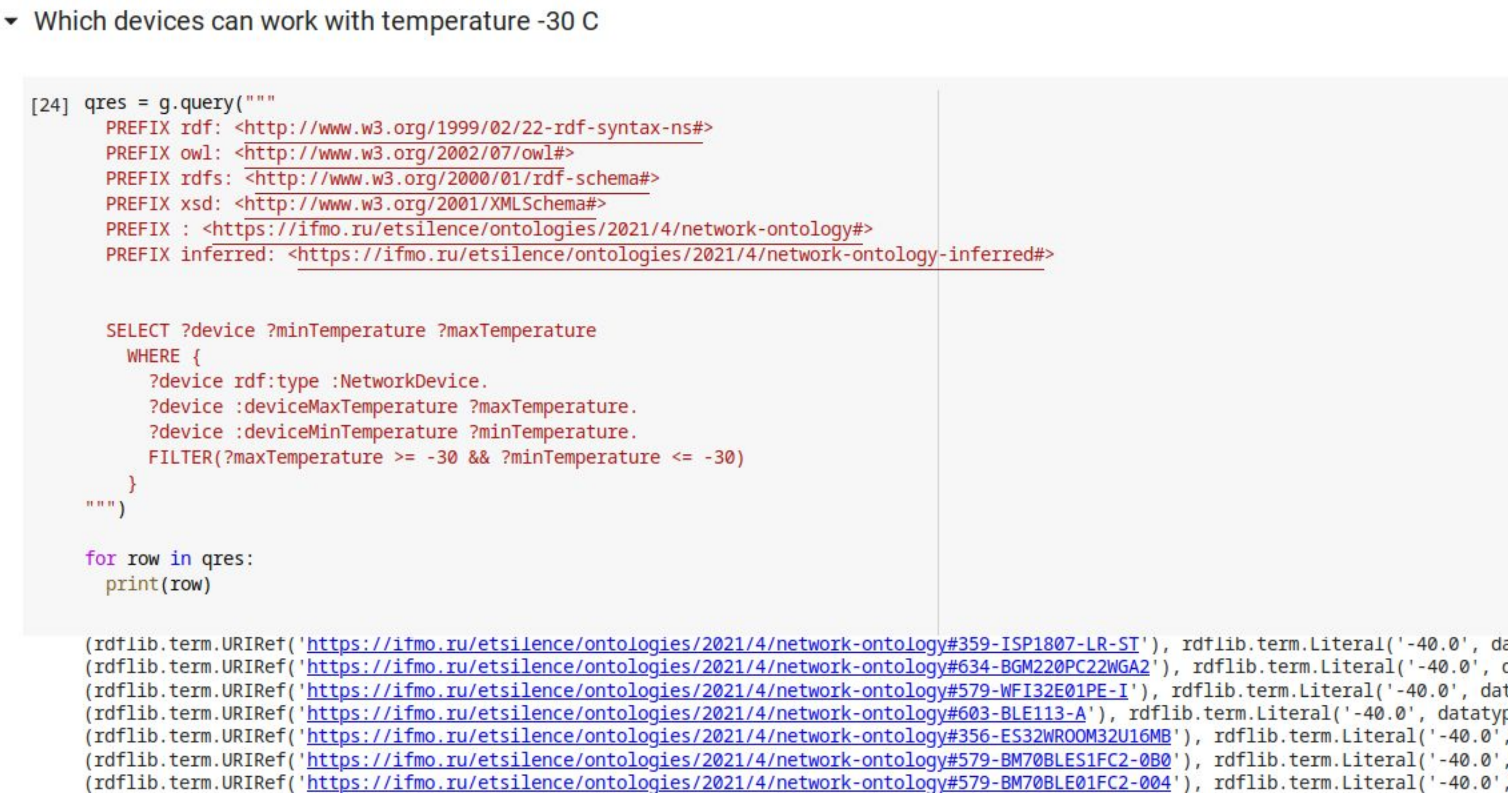


▾ Which devices can work on temperature +100 C

```
[23] qres = g.query("""
       PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
       PREFIX owl: <http://www.w3.org/2002/07/owl#>
       PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
       PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
       PREFIX : <https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#>
       PREFIX inferred: <https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology-inferred#>


       SELECT ?device ?minTemperature ?maxTemperature
         WHERE {
           ?device rdf:type :NetworkDevice.
           ?device :deviceMaxTemperature ?maxTemperature.
           ?device :deviceMinTemperature ?minTemperature.
           FILTER(?maxTemperature >= 100 && ?minTemperature <= 100)
         }
     """)

     for row in qres:
       print(row)
```

```
(rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#308-BM833E'), rdflib.term.Literal('-40.0', datatype=
(rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#634-BGM220PC22HNA2R'), rdflib.term.Literal('-40.0', d
(rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#634-WF200SD'), rdflib.term.Literal('-40.0', datatype=
(rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#377-NINA-B400-00B'), rdflib.term.Literal('-40.0', dat
(rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#634-BGM220PC22HNA2'), rdflib.term.Literal('-40.0', da
(rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#963-EYSPBNZUA'), rdflib.term.Literal('-40.0', datatyp
(rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#634-WFM200SS22XNN3'), rdflib.term.Literal('-40.0', da
(rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#634-WFM200S022XNA3'), rdflib.term.Literal('-40.0', da
```

# SPARQL

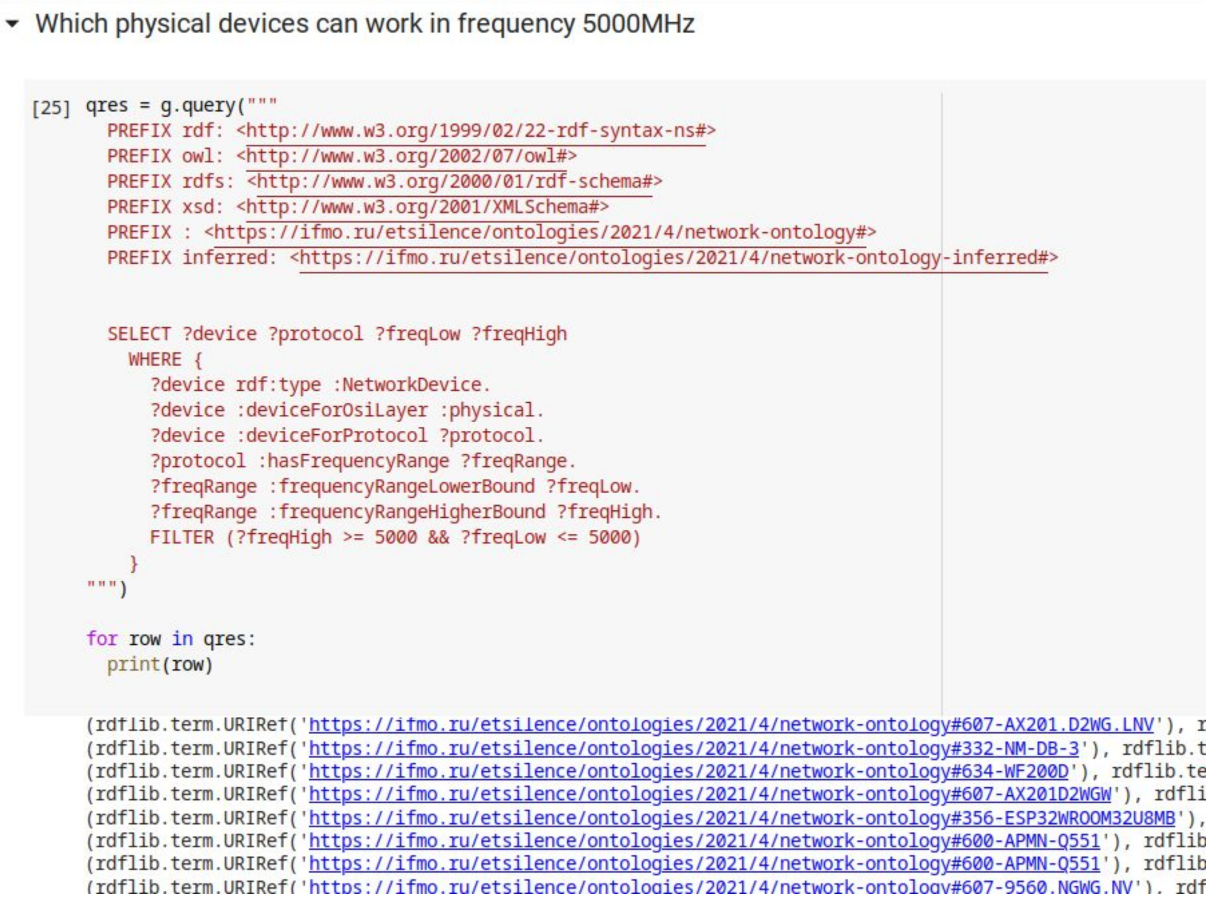


▾ Which devices can work with temperature -30 C

```
[24] qres = g.query("""
       PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
       PREFIX owl: <http://www.w3.org/2002/07/owl#>
       PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
       PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
       PREFIX : <https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#>
       PREFIX inferred: <https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology-inferred#>


       SELECT ?device ?minTemperature ?maxTemperature
         WHERE {
           ?device rdf:type :NetworkDevice.
           ?device :deviceMaxTemperature ?maxTemperature.
           ?device :deviceMinTemperature ?minTemperature.
           FILTER(?maxTemperature >= -30 && ?minTemperature <= -30)
         }
     """)

     for row in qres:
       print(row)
```

```
(rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#359-ISP1807-LR-ST'), rdflib.term.Literal('-40.0', da
(rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#634-BGM220PC22WGA2'), rdflib.term.Literal('-40.0', c
(rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#579-WFI32E01PE-I'), rdflib.term.Literal('-40.0', dat
(rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#603-BLE113-A'), rdflib.term.Literal('-40.0', datatyp
(rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#356-ES32WROOM32U16MB'), rdflib.term.Literal('-40.0',
(rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#579-BM70BLES1FC2-0B0'), rdflib.term.Literal('-40.0',
(rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#579-BM70BLE01FC2-004'), rdflib.term.Literal('-40.0',
```

# SPARQL

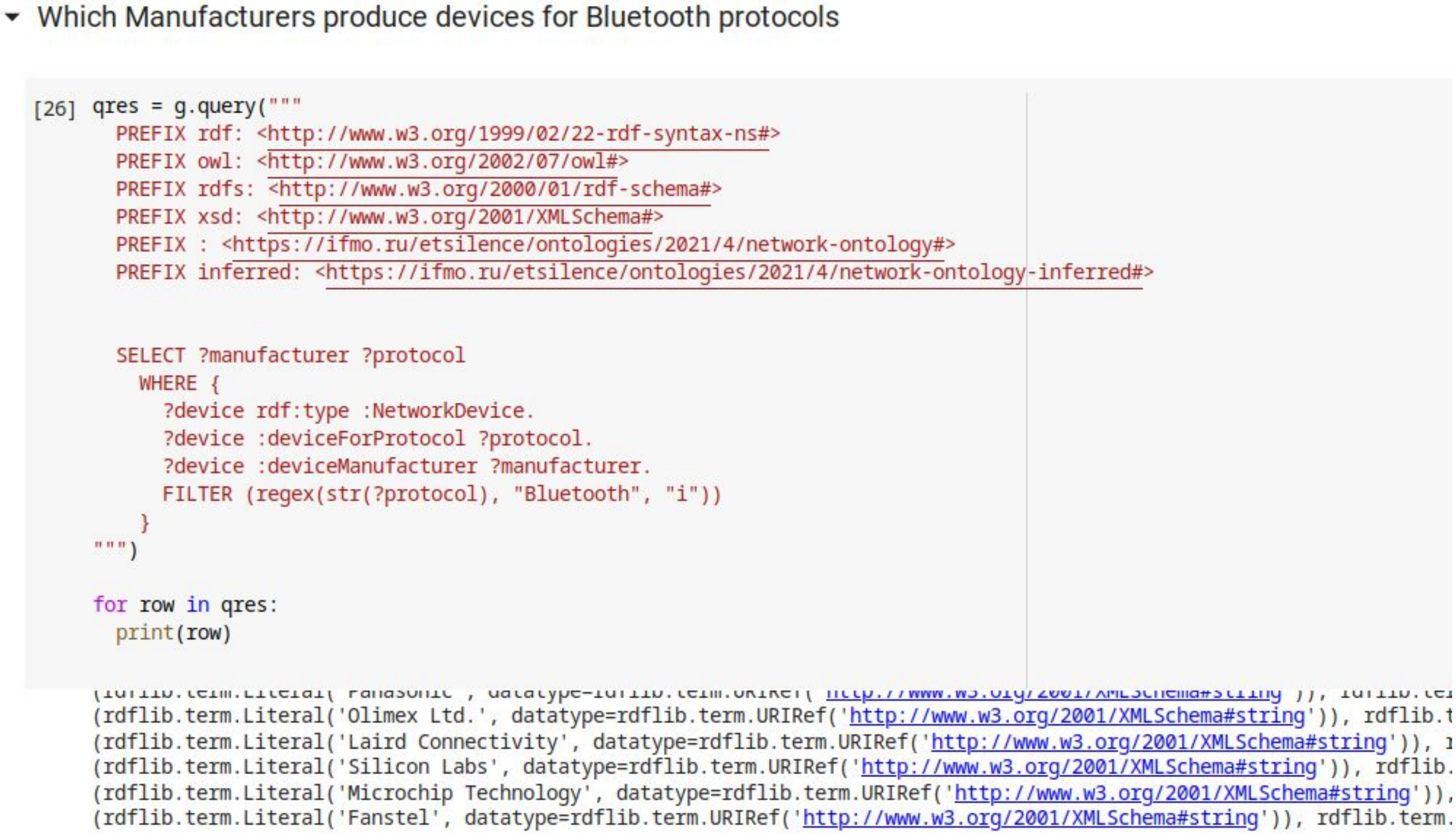


Which physical devices can work in frequency 5000MHz

```
[25] qres = g.query("""
       PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
       PREFIX owl: <http://www.w3.org/2002/07/owl#>
       PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
       PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
       PREFIX : <https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#>
       PREFIX inferred: <https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology-inferred#>


       SELECT ?device ?protocol ?freqLow ?freqHigh
         WHERE {
           ?device rdf:type :NetworkDevice.
           ?device :deviceForOsiLayer :physical.
           ?device :deviceForProtocol ?protocol.
           ?protocol :hasFrequencyRange ?freqRange.
           ?freqRange :frequencyRangeLowerBound ?freqLow.
           ?freqRange :frequencyRangeHigherBound ?freqHigh.
           FILTER (?freqHigh >= 5000 && ?freqLow <= 5000)
         }
     """)

     for row in qres:
       print(row)
```

```
(rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#607-AX201.D2WG.LNV'), r
(rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#332-NM-DB-3'), rdflib.t
(rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#634-WF200D'), rdflib.te
(rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#607-AX201D2WGW'), rdfli
(rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#356-ESP32WROOM32U8MB'),
(rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#600-APMN-Q551'), rdflib
(rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#600-APMN-Q551'), rdflib
(rdflib.term.URIRef('https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#607-9560.NGWG.NV'), rdf
```

# SPARQL

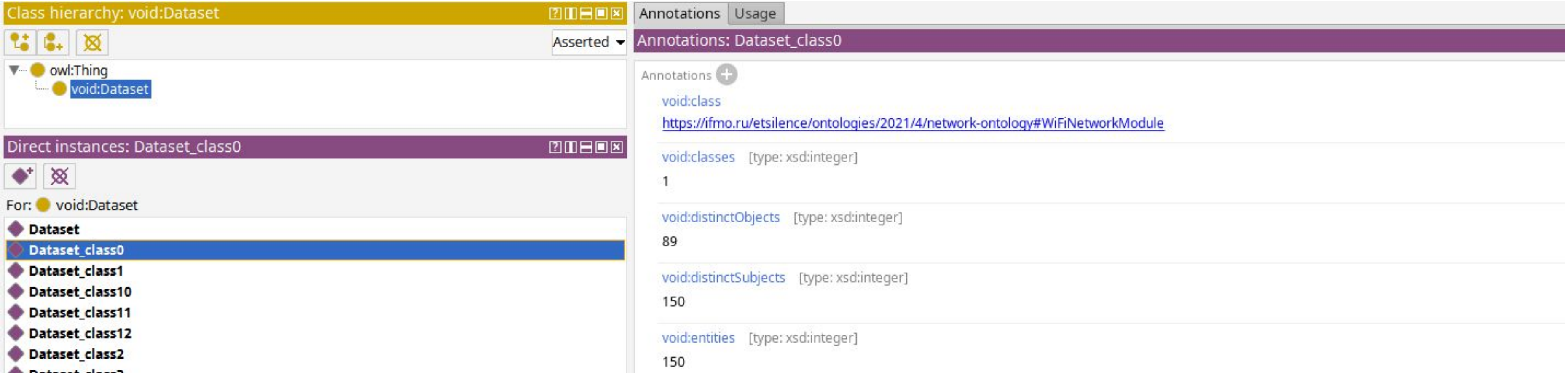


Which Manufacturers produce devices for Bluetooth protocols

```
[26] qres = g.query("""
       PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
       PREFIX owl: <http://www.w3.org/2002/07/owl#>
       PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
       PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
       PREFIX : <https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#>
       PREFIX inferred: <https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology-inferred#>


       SELECT ?manufacturer ?protocol
         WHERE {
           ?device rdf:type :NetworkDevice.
           ?device :deviceForProtocol ?protocol.
           ?device :deviceManufacturer ?manufacturer.
           FILTER (regex(str(?protocol), "Bluetooth", "i"))
         }
     """)

     for row in qres:
       print(row)
```

```
(rdflib.term.Literal('Panasonic', datatype=rdflib.term.URIRef('http://www.w3.org/2001/XMLSchema#string')), rdflib.ter
(rdflib.term.Literal('Olimex Ltd.', datatype=rdflib.term.URIRef('http://www.w3.org/2001/XMLSchema#string')), rdflib.t
(rdflib.term.Literal('Laird Connectivity', datatype=rdflib.term.URIRef('http://www.w3.org/2001/XMLSchema#string')), r
(rdflib.term.Literal('Silicon Labs', datatype=rdflib.term.URIRef('http://www.w3.org/2001/XMLSchema#string')), rdflib.
(rdflib.term.Literal('Microchip Technology', datatype=rdflib.term.URIRef('http://www.w3.org/2001/XMLSchema#string')),
(rdflib.term.Literal('Fanstel', datatype=rdflib.term.URIRef('http://www.w3.org/2001/XMLSchema#string')), rdflib.term.
```

# VoID



Class hierarchy: void:Dataset

- owl:Thing
  - void:Dataset

Direct instances: Dataset_class0

For: void:Dataset

- Dataset
- Dataset_class0
- Dataset_class1
- Dataset_class10
- Dataset_class11
- Dataset_class12
- Dataset_class2
- Dataset_class3

Annotations: Dataset_class0

Annotations

void:class
https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#WiFiNetworkModule

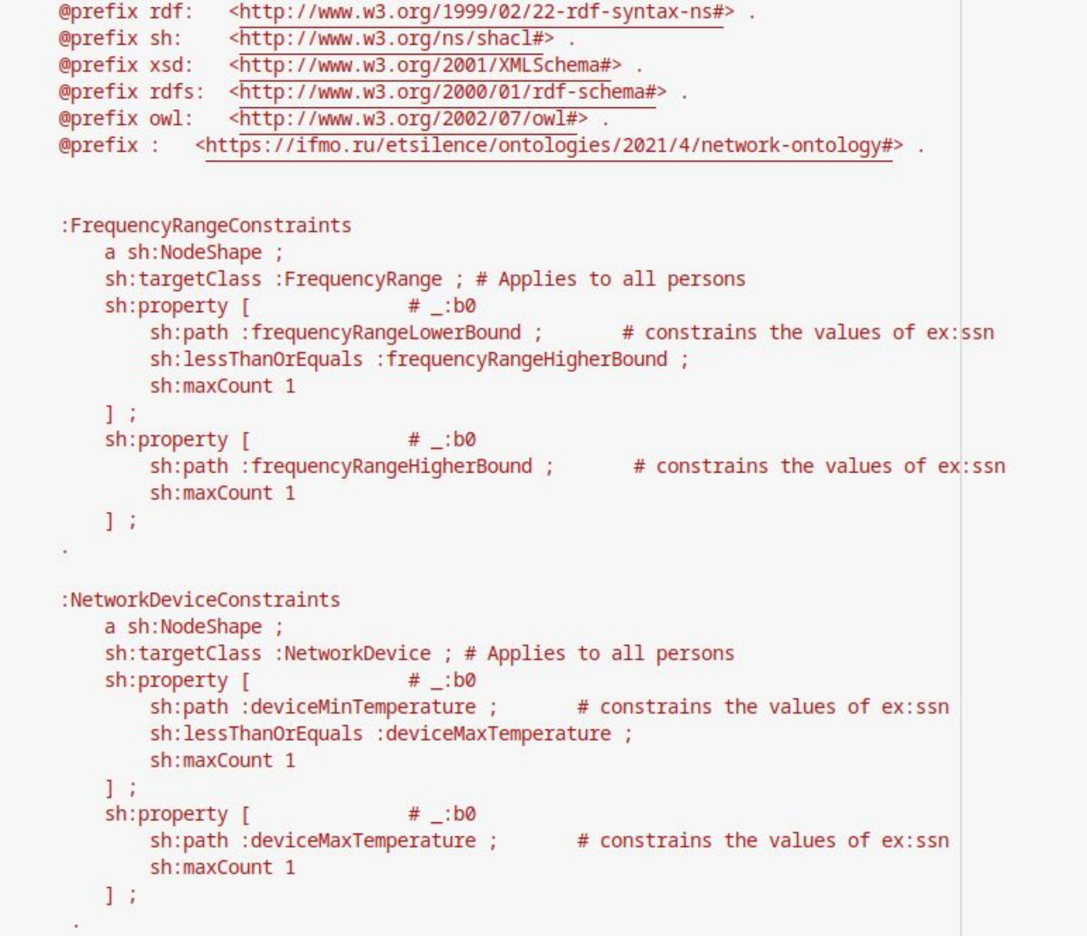
void:classes [type: xsd:integer]
1

void:distinctObjects [type: xsd:integer]
89

void:distinctSubjects [type: xsd:integer]
150

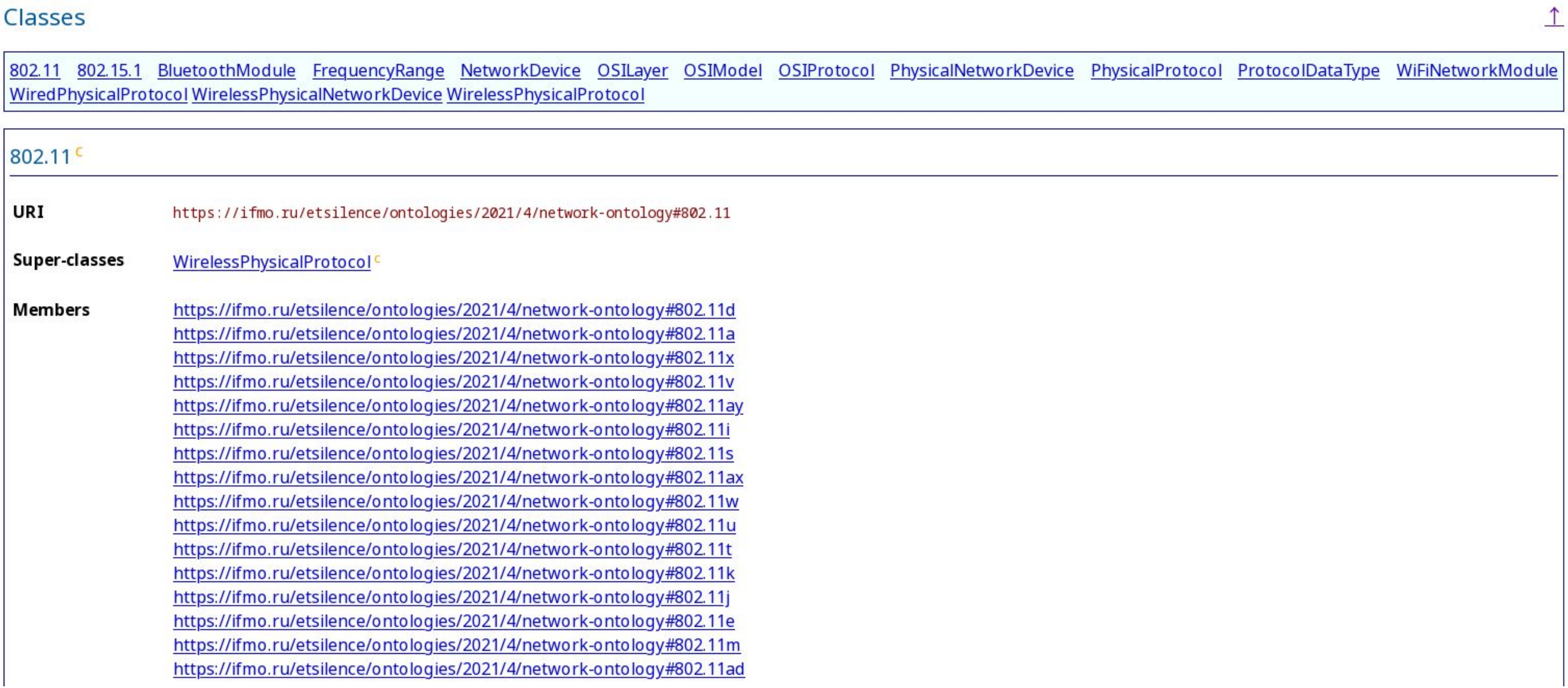
void:entities [type: xsd:integer]
150

# SHACL



```
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix sh:    <http://www.w3.org/ns/shacl#> .
@prefix xsd:   <http://www.w3.org/2001/XMLSchema#> .
@prefix rdfs:  <http://www.w3.org/2000/01/rdf-schema#> .
@prefix owl:   <http://www.w3.org/2002/07/owl#> .
@prefix :   <https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#> .


:FrequencyRangeConstraints
    a sh:NodeShape ;
    sh:targetClass :FrequencyRange ; # Applies to all persons
    sh:property [              # _:b0
        sh:path :frequencyRangeLowerBound ;       # constrains the values of ex:ssn
        sh:lessThanOrEquals :frequencyRangeHigherBound ;
        sh:maxCount 1
    ] ;
    sh:property [              # _:b0
        sh:path :frequencyRangeHigherBound ;       # constrains the values of ex:ssn
        sh:maxCount 1
    ] ;
.

:NetworkDeviceConstraints
    a sh:NodeShape ;
    sh:targetClass :NetworkDevice ; # Applies to all persons
    sh:property [              # _:b0
        sh:path :deviceMinTemperature ;       # constrains the values of ex:ssn
        sh:lessThanOrEquals :deviceMaxTemperature ;
        sh:maxCount 1
    ] ;
    sh:property [              # _:b0
        sh:path :deviceMaxTemperature ;       # constrains the values of ex:ssn
        sh:maxCount 1
    ] ;
 .
```

# Документация



## Classes

↑

802.11 802.15.1 BluetoothModule FrequencyRange NetworkDevice OSILayer OSIModel OSIProtocol PhysicalNetworkDevice PhysicalProtocol ProtocolDataType WiFiNetworkModule WiredPhysicalProtocol WirelessPhysicalNetworkDevice WirelessPhysicalProtocol

### 802.11 $^{c}$

**URI** `https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#802.11`

**Super-classes** WirelessPhysicalProtocol $^{c}$

**Members**
https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#802.11d
https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#802.11a
https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#802.11x
https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#802.11v
https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#802.11ay
https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#802.11i
https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#802.11s
https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#802.11ax
https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#802.11w
https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#802.11u
https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#802.11t
https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#802.11k
https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#802.11j
https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#802.11e
https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#802.11m
https://ifmo.ru/etsilence/ontologies/2021/4/network-ontology#802.11ad

# Возникшие проблемы

- Reasoning через python сгенерировал файл, который не интерпретируется парсером
- Данные с сайта Mouser.com можно выгружать только по одной странице поиска. Так как этих страниц много, то загрузка большого числа данных занимает много времени
- Данные с сайта Mouser.com находились в формате, который необходимо было дополнительно преобразовывать. Например протокол WiFi модуля 802.11a/b/g/n нужно было преобразовывать в набор протоколов 802.11a, 802.11b итд.
- Данные о частотных диапазонах протоколов приходилось находить и заполнять вручную, так как удобный для интерпретации формат не был найден

# Спасибо за внимание!

Хлопков Дмитрий
Университет ИТМО группа P41171
email: khlopkovdd@gmail.com