

Домашно бр. 1
Jena RDF API

А) Прашања

1. Како се изразува еден запис (еден факт) во RDF моделот?

Во RDF моделот, еден запис се претставува преку RDF тројка, која има три компоненти: субјект, предикат и објект. Субјектот го претставува ентитетот, предикатот е својството или релацијата која се изразува, додека објектот ја претставува вредноста или друг ентитет со кој е поврзан субјектот.

2. Кои различни синтакси за RDF моделот постојат? Изразете го следниот факт во неколку различни RDF синтакси: „БЕС се предава на ФИНКИ“. Користете го префиксот @prefix finki: <http://finki.ukim.mk/resource#> за URI вредностите на ентитетите и релацијата.

RDF моделот има повеќе синтакси за претставување, од кои најчесто се користат:

- RDF/XML
- Turtle (TTL)
- N-Triples
- JSON-LD
- RDFa

Пример за фактот „БЕС се предава на ФИНКИ“ во различни RDF синтакси со префикс @prefix finki: <http://finki.ukim.mk/resource#>:

- **RDF/XML:**

```
<rdf:RDF xmlns:finki="http://finki.ukim.mk/resource#">
  <rdf:Description rdf:about="http://finki.ukim.mk/resource#БЕС">
    <finki:се_предава_на rdf:resource="http://finki.ukim.mk/resource#ФИНКИ"/>
  </rdf:Description>
</rdf:RDF>
```

- **Turtle:**

```
@prefix finki: <http://finki.ukim.mk/resource#>.
finki:БЕС finki:се_предава_на finki:ФИНКИ.
```

- **N-Triples:**

```
<http://finki.ukim.mk/resource#БЕС>
<http://finki.ukim.mk/resource#се_предава_на>
<http://finki.ukim.mk/resource#ФИНКИ> .
```

3. За што се користи RDF Schema?

RDF Schema (RDFS) се користи за дефинирање на типови и релации во RDF моделот. Преку RDFS можеме да дефинираме класи, својства и нивни релации, со што се овозможува создавање на хиерархии и подобро разбирање на податоците.

4. Дефинирајте RDFS класи за „факултет“ и „предмет“, како и една релација која ги поврзува нив, „е предмет на“. Користете го префиксот од 2. за нивните URI вредности. Користете Turtle синтакса.

```
@prefix finki: <http://finki.ukim.mk/resource#> .
finki:Факултет rdf:type rdfs:Class .
finki:Предмет rdf:type rdfs:Class .
finki:еПредметНа rdf:type rdf:Property ;
rdfs:domain finki:Предмет ;
rdfs:range finki:Факултет .
```

Б) Практична задача

I. Креирање едноставен RDF граф

1. Креирајте нов Java проект во IDE по ваш избор. Вклучете ги во проектот сите .jar библиотеки од lib фолдерот од Jena. Jena преземете ја директно од [Jena сајтот](#).
2. Во main() методот на главната класа од проектот, креирајте основен Jena model, кој ќе го содржи RDF графот кој треба да го изградите во текот на вежбата.
3. Во моделот додадете нов ресурс, кој ќе ве репрезентира вас како личност. Како URI на ресурсот искористете URL адреса од некој ваш социјален профил (Facebook, Twitter, Instagram, TikTok, ...), кој уникатно ве идентификува.
4. Додадете својство на вашиот ресурс, кое ќе го репрезентира вашето целосно име. Искористете го својството 'vcard:fn'.
5. Додадете уште неколку својства по избор, кои ќе бидат од истата 'vcard' или пак од 'foaf' RDF шемата. Во моделот треба да имате минимум 10 RDF тројки. Притоа, внимавајте на тоа дали range вредноста на својството кое го додавате треба да биде литерал или друг објект.

```
import org.apache.jena.rdf.model.Model;
import org.apache.jena.rdf.model.ModelFactory;
import org.apache.jena.rdf.model.Resource;
import org.apache.jena.vocabulary.VCARD;

public class Main {
    public static void main(String[] args) {

        Model model = ModelFactory.createDefaultModel();

        Resource myProfile =
model.createResource("https://www.linkedin.com/in/bojan-ristov-827677272/");

        myProfile.addProperty(VCARD.FN, "Bojan Ristov");
        myProfile.addProperty(VCARD.BDAY, "2002-10-09");
        myProfile.addProperty(VCARD.EMAIL,
"bojan.ristov@students.finki.ukim.mk");
        myProfile.addProperty(VCARD.ADR, "Radovish");
        myProfile.addProperty(VCARD.Country, "Macedonia");

    }
}
```

```
1 import org.apache.jena.rdf.model.Model;
2 import org.apache.jena.rdf.model.ModelFactory;
3 import org.apache.jena.rdf.model.Resource;
4 import org.apache.jena.vocabulary.VCARD;
5
6 public class Main {
7     public static void main(String[] args) {
8
9         Model model = ModelFactory.createDefaultModel();
10
11         Resource myProfile = model.createResource( s: "https://www.linkedin.com/in/bojan-ristov-827677272/");
12
13         myProfile.addProperty(VCARD.FN, s: "Bojan Ristov");
14         myProfile.addProperty(VCARD.BDAY, s: "2002-10-09");
15         myProfile.addProperty(VCARD.EMAIL, s: "bojan.ristov@students.finki.ukim.mk");
16         myProfile.addProperty(VCARD.ADR, s: "Radovish");
17         myProfile.addProperty(VCARD.Country, s: "Macedonia");
18     }
19 }
20
21
```

II. Печатење на RDF граф

6. Со користење на `model.listStatements()` методот на моделот, изминете ги сите RDF записи (тројки) од графот и отпечатете ги во формат: “subject – predicate – object”. При печатењето, литералите отпечатете ги во наводници (“”). Печатењето нека биде во конзола, т.е. преку `System.out`.

Напомена: Пред да ги отпечатите RDF тројките, напишете на конзола “Printing with `model.listStatements()`.”.

7. Извршете ја програмата. Може да го извршите целиот проект или само класата во која го дефиниравте кодот до сега. Проверете дали излезот на конзола соодветствува со она што очекувате да се прикаже. Дали сите RDF тројки кои ги дефиниравте во кодот, ги гледате отпечатени?
8. Без да го бришете претходното печатење, додадете ново печатење на RDF тројките од моделот, со користење на `model.write()` методот. Притоа направете повеќе печатења, во следните RDF формати: RDF/XML, Pretty RDF/XML, N-Triples и Turtle.

Напомена: Пред секое од печатењата, напишете на конзола “Printing with `model.print()`, in *Turtle*.”, во зависност од конкретниот формат.

9. Извршете ја програмата. Може да го извршите целиот проект или само класата во која го дефиниравте кодот до сега. Проверете дали излезот на конзола соодветствува со она што очекувате да се прикаже. Кој од RDF форматите има најкратка (најкомпактна) содржина? Кој најлесно се „чита“ на прв поглед? Кој, пак, сметате дека најлесно би го испроцесирале во код, доколку го прочитате програмски од некаде?

```
System.out.println("Printing with model.write(), in RDF/XML format:");
model.write(System.out, "RDF/XML");

System.out.println("Printing with model.write(), in Turtle format:");
model.write(System.out, "TURTLE");

System.out.println("Printing with model.write(), in N-TRIPLES format:");
model.write(System.out, "N-TRIPLES");
```

```
Main.java x
1 import org.apache.jena.rdf.model.Model;
2 import org.apache.jena.rdf.model.ModelFactory;
3 import org.apache.jena.rdf.model.Resource;
4 import org.apache.jena.vocabulary.VCARD;
5
6 public class Main {
7     public static void main(String[] args) {
8
9         Model model = ModelFactory.createDefaultModel();
10
11         Resource myProfile = model.createResource(s: "https://www.linkedin.com/in/bojan-ristov-827677272/");
12
13         myProfile.addProperty(VCARD.FN, s: "Bojan Ristov");
14         myProfile.addProperty(VCARD.BDAY, s: "2002-10-09");
15         myProfile.addProperty(VCARD.EMAIL, s: "bojan.ristov@students.finki.ukim.mk");
16         myProfile.addProperty(VCARD.ADR, s: "Radovish");
17         myProfile.addProperty(VCARD.Country, s: "Macedonia");
18
19
20         System.out.println("Printing with model.write(), in RDF/XML format:");
21         model.write(System.out, s: "RDF/XML");
22
23         System.out.println("Printing with model.write(), in Turtle format:");
24         model.write(System.out, s: "TURTLE");
25
26         System.out.println("Printing with model.write(), in N-TRIPLES format:");
27         model.write(System.out, s: "N-TRIPLES");
28     }
29 }
```

Output:

Printing with model.write(), in RDF/XML format:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:vcard="http://www.w3.org/2001/vcard-rdf/3.0#">
  <rdf:Description rdf:about="https://www.linkedin.com/in/bojan-ristov-827677272/">
    <vcard:Country>Macedonia</vcard:Country>
    <vcard:ADR>Radovish</vcard:ADR>
    <vcard:EMAIL>bojan.ristov@students.finki.ukim.mk</vcard:EMAIL>
    <vcard:BDAY>2002-10-09</vcard:BDAY>
    <vcard:FN>Bojan Ristov</vcard:FN>
  </rdf:Description>
</rdf:RDF>
```

Printing with model.write(), in Turtle format:

```
<https://www.linkedin.com/in/bojan-ristov-827677272/>
  <http://www.w3.org/2001/vcard-rdf/3.0#ADR>
    "Radovish";
  <http://www.w3.org/2001/vcard-rdf/3.0#BDAY>
    "2002-10-09";
  <http://www.w3.org/2001/vcard-rdf/3.0#Country>
    "Macedonia";
  <http://www.w3.org/2001/vcard-rdf/3.0#EMAIL>
    "bojan.ristov@students.finki.ukim.mk";
  <http://www.w3.org/2001/vcard-rdf/3.0#FN>
    "Bojan Ristov" .
```

Printing with model.write(), in N-TRIPLES format:

```
<https://www.linkedin.com/in/bojan-ristov-827677272/>
rdf/3.0#Country> "Macedonia" .
<https://www.linkedin.com/in/bojan-ristov-827677272/>
rdf/3.0#ADR> "Radovish" .
<https://www.linkedin.com/in/bojan-ristov-827677272/>
rdf/3.0#EMAIL> "bojan.ristov@students.finki.ukim.mk" .
<https://www.linkedin.com/in/bojan-ristov-827677272/>
rdf/3.0#BDAY> "2002-10-09" .
<https://www.linkedin.com/in/bojan-ristov-827677272/>
rdf/3.0#FN> "Bojan Ristov" .
```

```
<http://www.w3.org/2001/vcard-
<http://www.w3.org/2001/vcard-
<http://www.w3.org/2001/vcard-
<http://www.w3.org/2001/vcard-
<http://www.w3.org/2001/vcard-
```

Process finished with exit code 0

Printing with model.write(), in RDF/XML format:

```
<rdf:RDF
  xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:vcard="http://www.w3.org/2001/vcard-rdf/3.0#"
  <rdf:Description rdf:about="https://www.linkedin.com/in/bojan-ristov-827677272/">
    <vcard:Country>Macedonia</vcard:Country>
    <vcard:ADR>Radovish</vcard:ADR>
    <vcard:EMAIL>bojan.ristov@students.finki.ukim.mk</vcard:EMAIL>
    <vcard:BDAY>2002-10-09</vcard:BDAY>
    <vcard:FN>Bojan Ristov</vcard:FN>
  </rdf:Description>
</rdf:RDF>
```

Printing with model.write(), in Turtle format:

```
<https://www.linkedin.com/in/bojan-ristov-827677272/>
  <http://www.w3.org/2001/vcard-rdf/3.0#ADR>
    "Radovish";
  <http://www.w3.org/2001/vcard-rdf/3.0#BDAY>
    "2002-10-09";
  <http://www.w3.org/2001/vcard-rdf/3.0#Country>
    "Macedonia";
  <http://www.w3.org/2001/vcard-rdf/3.0#EMAIL>
    "bojan.ristov@students.finki.ukim.mk";
  <http://www.w3.org/2001/vcard-rdf/3.0#FN>
    "Bojan Ristov" .
```

Printing with model.write(), in N-TRIPLES format:

```
<https://www.linkedin.com/in/bojan-ristov-827677272/> <http://www.w3.org/2001/vcard-rdf/3.0#Country> "Macedonia" .
<https://www.linkedin.com/in/bojan-ristov-827677272/> <http://www.w3.org/2001/vcard-rdf/3.0#ADR> "Radovish" .
<https://www.linkedin.com/in/bojan-ristov-827677272/> <http://www.w3.org/2001/vcard-rdf/3.0#EMAIL> "bojan.ristov@students.finki.ukim.mk" .
<https://www.linkedin.com/in/bojan-ristov-827677272/> <http://www.w3.org/2001/vcard-rdf/3.0#BDAY> "2002-10-09" .
<https://www.linkedin.com/in/bojan-ristov-827677272/> <http://www.w3.org/2001/vcard-rdf/3.0#FN> "Bojan Ristov" .|
```

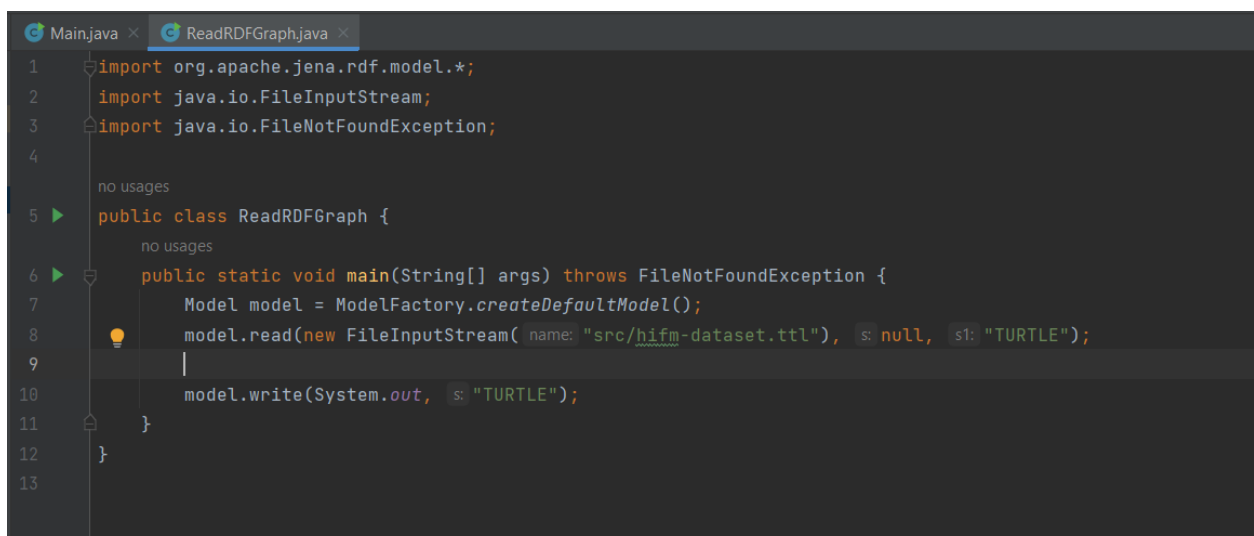
III. Читање на RDF граф

10. Креирајте нова Java класа во проектот, во која ќе додадете и `main()` метод.
11. Ископирајте еден од излезите од претходните задачи (вашиот RDF граф во некоја од RDF синтаксите) и ставете го во текстуален фајл, кој ќе го снимите локално, под произволно име и соодветна наставка: `.xml` за RDF/XML и Pretty RDF/XML, `.ttl` за Turtle, `.nt` за N-Triples и `n3` за N3.
12. Во `main()` методот на новата класа креирајте нов модел и со користење на `model.read()` вчитајте го RDF графот од датотеката креирана во претходниот чекор.
Напомена: Искористете го третиот параметар на `model.read()` кој го означува RDF форматот на датотеката која ја читате – има исти вредности како `model.write()` при запишување, односно “RDF/XML”, “RDF/XML-ABBREV”, “TTL”, “N-TRIPLES”, итн.
13. Напишете код за печатење на моделот (графот), за да видите дали успешно е прочитан.
14. Извршете ја програмата. Може да го извршите целиот проект или само класата во која го дефиниравте кодот до сега. Проверете дали излезот на конзола соодветствува со она што очекувате да се прикаже.

```
import org.apache.jena.rdf.model.*;
import java.io.FileInputStream;
import java.io.FileNotFoundException;

public class ReadRDFGraph {
    public static void main(String[] args) throws FileNotFoundException {
        Model model = ModelFactory.createDefaultModel();
        model.read(new FileInputStream("src/hifm-dataset.ttl"), null,
"TURTLE");

        model.write(System.out, "TURTLE");
    }
}
```



```

ReadRDFGraph x
989673> , <http://purl.org/net/hifm/data#994707> , <http://purl.org/net/hifm/data#984892> , <http://purl.org/net/hifm/data#985082> , <http://purl.org/net/hifm/data#989134>
hifm-ont:strength "20mg";
drugbank:atcCode "C10AA";
drugbank:brandName "ATOLIP фимм обл.табл. 30 x 20mg";
drugbank:genericName "Atorvastatin" .

<http://purl.org/net/hifm/data#987123>
rdf:type drugbank:drugs , hifm-ont:Drug;
rdfs:label "Salmeterol, fluticasone";
hifm-ont:dosageForm "Прашок за инхалирање";
hifm-ont:id 987123;
hifm-ont:manufacturer "GSK";
hifm-ont:packaging 1;
hifm-ont:refPriceNoVAT 1694.29;
hifm-ont:refPriceWithVAT 1779.0;
hifm-ont:similarTo <http://purl.org/net/hifm/data#988731> , <http://purl.org/net/hifm/data#987131>;
hifm-ont:strength "50mcg+250mcg";
drugbank:atcCode "R03AK06";
drugbank:brandName "SERETIDE DISKUS Прашок за инхалирање 50mcg+250mcg (60 Dozi)";
drugbank:genericName "Salmeterol, fluticasone" .

<http://purl.org/net/hifm/data#979392>
rdf:type drugbank:drugs , hifm-ont:Drug;
rdfs:label "Brinzolamide";
rdfs:seeAlso ns5:DB01194;
hifm-ont:dosageForm "Капки за очи";
hifm-ont:id 979392;
hifm-ont:manufacturer "ALCON";
hifm-ont:packaging 1;
hifm-ont:refPriceNoVAT 445.71;
hifm-ont:refPriceWithVAT 468.0;
hifm-ont:strength "10 mg/ml";
drugbank:atcCode "S01EC04";
drugbank:brandName "AZOPT капки за очи 10mg/ml (1%) (5ml)";
drugbank:genericName "Brinzolamide" .

```

Output:

```

<http://purl.org/net/hifm/data#979392>
  rdf:type      drugbank:drugs , hifm-ont:Drug;
  rdfs:label    "Brinzolamide";
  rdfs:seeAlso  ns5:DB01194;
  hifm-ont:dosageForm "Капки за очи";
  hifm-ont:id    979392;
  hifm-ont:manufacturer "ALCON";
  hifm-ont:packaging 1;
  hifm-ont:refPriceNoVAT 445.71;
  hifm-ont:refPriceWithVAT 468.0;
  hifm-ont:strength "10 mg/ml";
  drugbank:atcCode "S01EC04";
  drugbank:brandName "AZOPT капки за очи 10mg/ml (1%) (5ml)";
  drugbank:genericName "Brinzolamide" .

<http://purl.org/net/hifm/data#968323>
  rdf:type      hifm-ont:Drug , drugbank:drugs;
  rdfs:label    "Sulfamethoxazole+trimethoprim";
  hifm-ont:dosageForm "Таблети";
  hifm-ont:id    968323;
  hifm-ont:manufacturer "ЈАКА 80";
  hifm-ont:packaging 20;
  hifm-ont:refPriceNoVAT 20.95;
  hifm-ont:refPriceWithVAT 22.0;
  hifm-ont:similarTo <http://purl.org/net/hifm/data#993778> ,
  <http://purl.org/net/hifm/data#980099> , <http://purl.org/net/hifm/data#23833> ,
  <http://purl.org/net/hifm/data#980269> , <http://purl.org/net/hifm/data#968315> ,
  <http://purl.org/net/hifm/data#963364> , <http://purl.org/net/hifm/data#963356>;
  hifm-ont:strength "(100+20)mg";

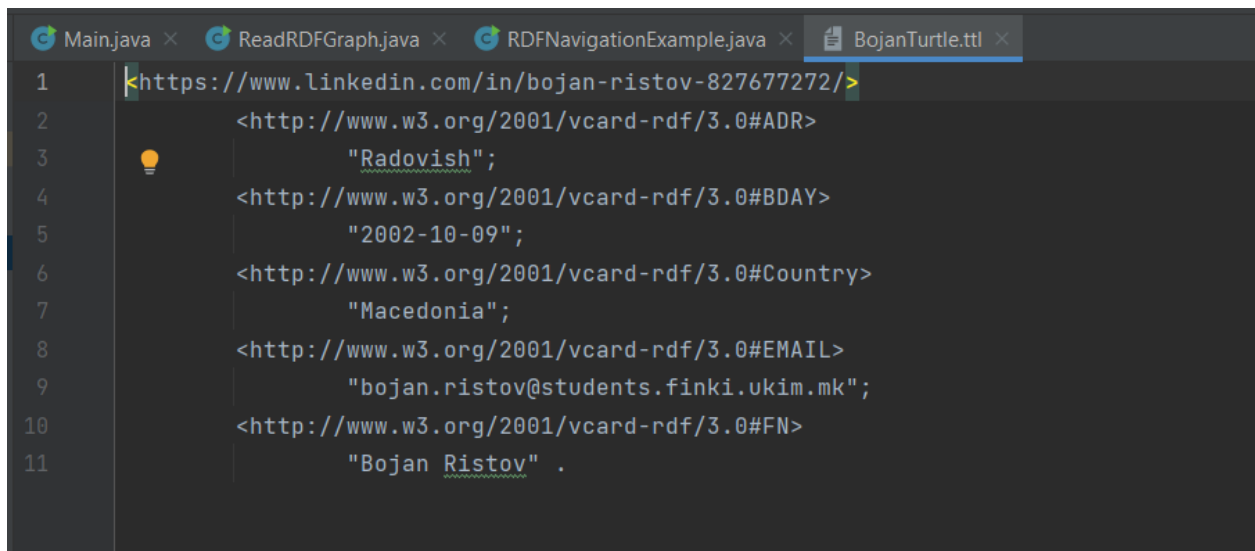
```

...

IV. Навигација низ RDF граф

15. Откако ќе го вчитате графот од датотека во претходниот дел од вежбата, додадете код кој ќе го селектира ресурсот од графот кој ве репрезентира вас.
16. Преку селектираниот ресурс, прочитајте ја вредноста на дел од релациите (целосно име, име, презиме, итн.), во зависност од тоа што сте креирале како RDF тројки на почетокот од вежбата.
Напомена: Внимавајте како пристапувате до вредностите кои се ресурси, а како до вредностите кои се литерали. Постои ли разлика во начинот на пристап?
17. Извршете ја програмата. Може да го извршите целиот проект или само класата во која го дефиниравте кодот до сега. Проверете дали излезот на конзола соодветствува со она што очекувате да се прикаже.

```
<https://www.linkedin.com/in/bojan-ristov-827677272/>
  <http://www.w3.org/2001/vcard-rdf/3.0#ADR>
    "Radovish";
  <http://www.w3.org/2001/vcard-rdf/3.0#BDAY>
    "2002-10-09";
  <http://www.w3.org/2001/vcard-rdf/3.0#Country>
    "Macedonia";
  <http://www.w3.org/2001/vcard-rdf/3.0#EMAIL>
    "bojan.ristov@students.finki.ukim.mk";
  <http://www.w3.org/2001/vcard-rdf/3.0#FN>
    "Bojan Ristov" .
```



```
1 <https://www.linkedin.com/in/bojan-ristov-827677272/>
2   <http://www.w3.org/2001/vcard-rdf/3.0#ADR>
3     "Radovish";
4   <http://www.w3.org/2001/vcard-rdf/3.0#BDAY>
5     "2002-10-09";
6   <http://www.w3.org/2001/vcard-rdf/3.0#Country>
7     "Macedonia";
8   <http://www.w3.org/2001/vcard-rdf/3.0#EMAIL>
9     "bojan.ristov@students.finki.ukim.mk";
10  <http://www.w3.org/2001/vcard-rdf/3.0#FN>
11    "Bojan Ristov" .
```

```
import org.apache.jena.rdf.model.*;

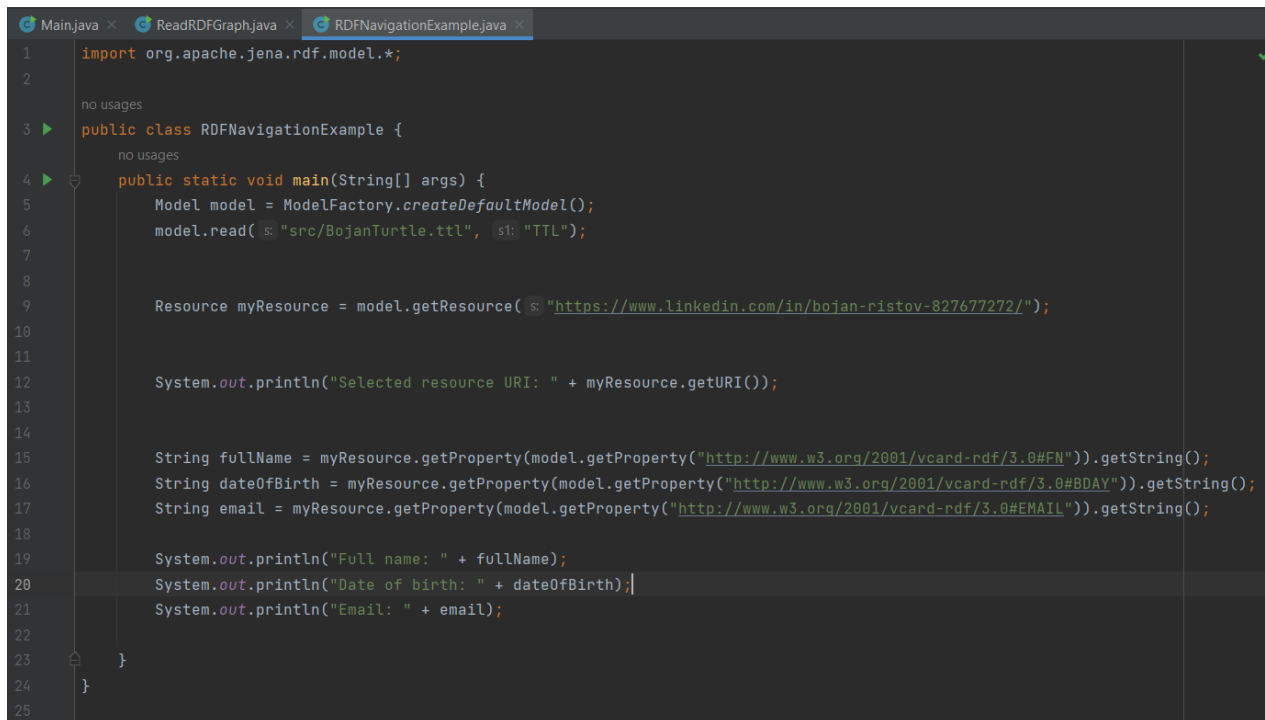
public class RDFNavigationExample {
    public static void main(String[] args) {
        Model model = ModelFactory.createDefaultModel();
        model.read("src/BojanTurtle.ttl", "TTL");

        Resource myResource =
model.getResource("https://www.linkedin.com/in/bojan-ristov-827677272/");

        System.out.println("Selected resource URI: " + myResource.getURI());

        String fullName =
myResource.getProperty(model.getProperty("http://www.w3.org/2001/vcard-
rdf/3.0#FN")).getString();
        String dateOfBirth =
myResource.getProperty(model.getProperty("http://www.w3.org/2001/vcard-
rdf/3.0#BDAY")).getString();
        String email =
myResource.getProperty(model.getProperty("http://www.w3.org/2001/vcard-
rdf/3.0#EMAIL")).getString();

        System.out.println("Full name: " + fullName);
        System.out.println("Date of birth: " + dateOfBirth);
        System.out.println("Email: " + email);
    }
}
```



The screenshot shows an IDE window with three tabs: Main.java, ReadRDFGraph.java, and RDFNavigationExample.java. The active tab is RDFNavigationExample.java, which displays the same Java code as the previous block. The code is numbered from 1 to 25. Annotations include "no usages" for the class and the main method. Line 20 is highlighted with a mouse cursor. The code is as follows:

```
1 import org.apache.jena.rdf.model.*;
2
3 public class RDFNavigationExample {
4     public static void main(String[] args) {
5         Model model = ModelFactory.createDefaultModel();
6         model.read("src/BojanTurtle.ttl", "TTL");
7
8         Resource myResource = model.getResource("https://www.linkedin.com/in/bojan-ristov-827677272/");
9
10        System.out.println("Selected resource URI: " + myResource.getURI());
11
12        String fullName = myResource.getProperty(model.getProperty("http://www.w3.org/2001/vcard-rdf/3.0#FN")).getString();
13        String dateOfBirth = myResource.getProperty(model.getProperty("http://www.w3.org/2001/vcard-rdf/3.0#BDAY")).getString();
14        String email = myResource.getProperty(model.getProperty("http://www.w3.org/2001/vcard-rdf/3.0#EMAIL")).getString();
15
16        System.out.println("Full name: " + fullName);
17        System.out.println("Date of birth: " + dateOfBirth);
18        System.out.println("Email: " + email);
19    }
20 }
21
22
23
24
25
```



```
RDFNavigationExample x
C:\Users\pc\.jdk\openjdk-19.0.1\bin\java.exe ...
Selected resource URI: https://www.linkedin.com/in/bojan-ristov-827677272/
Full name: Bojan Ristov
Date of birth: 2002-10-09
Email: bojan.ristov@students.finki.ukim.mk
Process finished with exit code 0
```

Output:

Selected resource URI: <https://www.linkedin.com/in/bojan-ristov-827677272/>
Full name: Bojan Ristov
Date of birth: 2002-10-09
Email: bojan.ristov@students.finki.ukim.mk

Process finished with exit code 0

V. Извлекување податоци од RDF граф

18. Креирајте нова Java класа во проектот, во која ќе додадете и `main()` метод.
19. Преземете ја датотеката “`hifm-dataset.ttl`” од Courses и снимете ја локално.
20. Во `main()` методот на новата класа напишете код со кој ќе ја прочитате содржината на оваа датотека. Внимавајте третиот параметар на `model.read()` да го поставите за вчитување на Turtle содржина.
21. Проучете ја содржината на “`hifm-dataset.ttl`” датотеката. Станува збор за податочно множество кое содржи лекови од Фондот за здравство на РМ. За секој од лековите имаме тип (`hifm-ont:Drug` и `drugbank:drugs`), име (`rdfs:label`, `drugbank:brandName` и `drugbank:genericName`), цена (`hifm-ont:refPriceWithVAT`), релации кон други локални (`hifm-ont:similarTo`) и светски лекови (`rdfs:seeAlso`), итн.
22. Врз база на наученото од досегашниот тек на вежбата, излистајте ги имињата на сите лекови кои се наоѓаат во графот (моделот) (една од трите релации за име е доволна), по азбучен редослед.
23. Одберете еден лек од графот (моделот) и за него излистајте ги сите релации и вредности.

24. Одберете еден лек од графот (моделот) и за него излистајте ги имињата на сите лекови кои имаат иста функција како и тој, т.е. лекови со кои тој е во релација 'hifm-ont:similarTo'. Форматирајте го печатењето за да е јасно видливо за што станува збор.
25. Одберете еден лек од графот (моделот) и за него најпрвин излистајте ја неговата цена (hifm-ont:refPriceWithVAT), а потоа излистајте ги и имињата и цените на лековите кои ја имаат истата функција како и тој (hifm-ont:similarTo). Форматирајте го печатењето за да е јасно видливо за што станува збор.
26. Извршете ја програмата. Може да го извршите целиот проект или само класата во која го дефиниравте кодот до сега. Проверете дали излезот на конзола соодветствува со она што очекувате да се прикаже.

Напомена: Доколку успеавте да ги завршите задачите под точка 22, 23, 24 и 25, практично напишавте код кој може да биде основа за една мобилна, веб или десктоп апликација за лекови: на корисникот му се претставуваат сите лекови (22), може да одбере некој од нив и да му се отвори приказ со сите детали за лекот (23), да ги види алтернативните лекови со иста функција кои може да ги купи наместо селектираниот (24) и да ги спореди нивните цени (25) со цел да го избере најевтиниот од таа група лекови со исто дејство.

Вчитување на RDF граф од датотека и листање на имињата на сите лекови во графот по азбучен редослед

```
import org.apache.jena.rdf.model.Model;
import org.apache.jena.rdf.model.ModelFactory;
import org.apache.jena.vocabulary.RDFS;
import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class HIFMDatasetProcessor {
    public static void main(String[] args) {
        Model model = ModelFactory.createDefaultModel();
        model.read("src/hifm-dataset.ttl", "TTL");

        System.out.println("Model loaded successfully.");

        List<String> drugNames = new ArrayList<>();

        model.listSubjectsWithProperty(RDFS.label).forEachRemaining(drug -> {
            String name = drug.getProperty(RDFS.label).getString();
            drugNames.add(name);
        });

        Collections.sort(drugNames);

        drugNames.forEach(System.out::println);
    }
}
```

```

Main.java × ReadRDFGraph.java × RDFNavigationExample.java × HIFMDataSetProcessor.java ×
1  import org.apache.jena.rdf.model.Model;
2  import org.apache.jena.rdf.model.ModelFactory;
3  import org.apache.jena.vocabulary.RDFS;
4  import java.util.ArrayList;
5  import java.util.Collections;
6  import java.util.List;
7
8  no usages
9  public class HIFMDataSetProcessor {
10     no usages
11     public static void main(String[] args) {
12         Model model = ModelFactory.createDefaultModel();
13         model.read("src/hifm-dataset.ttl", "TTL");
14
15         System.out.println("Model loaded successfully.");
16
17         List<String> drugNames = new ArrayList<>();
18
19         model.listSubjectsWithProperty(RDFS.Label).forEachRemaining(drug -> {
20             String name = drug.getProperty(RDFS.Label).getString();
21             drugNames.add(name);
22         });
23
24         Collections.sort(drugNames);
25
26         drugNames.forEach(System.out::println);
27     }
28 }
29

```

Output:

Model loaded successfully.
 Acenocoumarol
 Aciclovir
 Acidum Acetylsalicylicum
 Albendazole
 Alendronat sodium, colecalciferol
 Alendronic acid
 Allopurinol
 Alprazolam
 Amilase +Lipase+Protease/Pankreatin
 Aminophylline
 Amiodarone
 Amitriptyline
 Amlodipine
 Amoxicillin
 Amoxicillin +clavulanic acid
 Atenolol

...

Детали за одбран лек

```
import org.apache.jena.rdf.model.Model;
import org.apache.jena.rdf.model.ModelFactory;
import org.apache.jena.rdf.model.Resource;
import org.apache.jena.vocabulary.RDFS;

import java.util.ArrayList;
import java.util.Collections;
import java.util.List;

public class HIFMDatasetProcessor {
    public static void main(String[] args) {
        Model model = ModelFactory.createDefaultModel();
        model.read("src/hifm-dataset.ttl", "TTL");

        System.out.println("Model loaded successfully.");

        List<String> drugNames = new ArrayList<>();

        model.listSubjectsWithProperty(RDFS.label).forEachRemaining(drug -> {
            String name = drug.getProperty(RDFS.label).getString();
            drugNames.add(name);
        });

        Collections.sort(drugNames);

        //drugNames.forEach(System.out::println);

        Resource selectedDrug =
model.getResource("http://purl.org/net/hifm/data#986712");

        selectedDrug.listProperties().forEachRemaining(statement -> {
            System.out.println("Property: " +
statement.getPredicate().getLocalName() +
            " - Value: " + statement.getObject().toString());
        });
    }
}
```

```
27
28
29     Resource selectedDrug = model.getResource("http://purl.org/net/hifm/data#986712");
30
31     selectedDrug.listProperties().forEachRemaining(statement -> {
32         System.out.println("Property: " + statement.getPredicate().getLocalName() +
33             " - Value: " + statement.getObject().toString());
34     });
35
36 }
```

```

HIFMDataSetProcessor x
C:\Users\pc\.jdk\openjdk-19.0.1\bin\java.exe ...
Model loaded successfully.
Property: refPriceWithVAT - Value: "97.0"^^xsd:decimal
Property: similarTo - Value: http://purl.org/net/hifm/data#975354
Property: strength - Value: 40mg
Property: brandName - Value: ULKOBOS капс. 14 x 40mg
Property: similarTo - Value: http://purl.org/net/hifm/data#993662
Property: similarTo - Value: http://purl.org/net/hifm/data#82481
Property: similarTo - Value: http://purl.org/net/hifm/data#75949
Property: manufacturer - Value: BOSNALIJEK
Property: refPriceNoVAT - Value: "92.38"^^xsd:decimal
Property: type - Value: http://purl.org/net/hifm/ontology#Drug
Property: packaging - Value: "14"^^xsd:integer
Property: seeAlso - Value: http://wifo5-04.informatik.uni-mannheim.de/drugbank/resource/drugs/DB00736
Property: id - Value: "986712"^^xsd:integer
Property: dosageForm - Value: Капсули
Property: type - Value: http://wifo5-04.informatik.uni-mannheim.de/drugbank/resource/drugbank/drugs
Property: seeAlso - Value: http://wifo5-04.informatik.uni-mannheim.de/drugbank/resource/drugs/DB00338
Property: similarTo - Value: http://purl.org/net/hifm/data#950076
Property: similarTo - Value: http://purl.org/net/hifm/data#975397
Property: similarTo - Value: http://purl.org/net/hifm/data#83496
Property: label - Value: Omeprazole
Property: genericName - Value: Omeprazole
Property: atcCode - Value: A02BC01

Process finished with exit code 0

```

Output:

Property: refPriceWithVAT - Value: "97.0"^^xsd:decimal
 Property: similarTo - Value: <http://purl.org/net/hifm/data#975354>
 Property: strength - Value: 40mg
 Property: brandName - Value: ULKOBOS капс. 14 x 40mg
 Property: similarTo - Value: <http://purl.org/net/hifm/data#993662>
 Property: similarTo - Value: <http://purl.org/net/hifm/data#82481>
 Property: similarTo - Value: <http://purl.org/net/hifm/data#75949>
 Property: manufacturer - Value: BOSNALIJEK
 Property: refPriceNoVAT - Value: "92.38"^^xsd:decimal
 Property: type - Value: <http://purl.org/net/hifm/ontology#Drug>
 Property: packaging - Value: "14"^^xsd:integer
 Property: seeAlso - Value: <http://wifo5-04.informatik.uni-mannheim.de/drugbank/resource/drugs/DB00736>
 Property: id - Value: "986712"^^xsd:integer
 Property: dosageForm - Value: Капсули
 Property: type - Value: <http://wifo5-04.informatik.uni-mannheim.de/drugbank/resource/drugbank/drugs>
 Property: seeAlso - Value: <http://wifo5-04.informatik.uni-mannheim.de/drugbank/resource/drugs/DB00338>
 Property: similarTo - Value: <http://purl.org/net/hifm/data#950076>
 Property: similarTo - Value: <http://purl.org/net/hifm/data#975397>
 Property: similarTo - Value: <http://purl.org/net/hifm/data#83496>
 Property: label - Value: Omeprazole
 Property: genericName - Value: Omeprazole
 Property: atcCode - Value: A02BC01

Лекови со иста функција и имиња и цени на лекови со иста функција

```
Resource drug = model.getResource("http://purl.org/net/hifm/data#986712");

Property similarTo =
model.getProperty("http://purl.org/net/hifm/data#986712/hifm-ont:similarTo");

drug.listProperties(similarTo).forEachRemaining(statement -> {
    Resource similarDrug = statement.getObject().asResource();
    String drugName = similarDrug.getProperty(RDFS.label).getString();
    System.out.println("Similar drug: " + drugName);
});

Property priceProperty =
model.getProperty("http://purl.org/net/hifm/data#987123/hifm-
ont:refPriceWithVAT");

String drugPrice = drug.getProperty(priceProperty).getString();
System.out.println("Price of selected drug: " + drugPrice);

drug.listProperties(similarTo).forEachRemaining(statement -> {
    Resource similarDrug = statement.getObject().asResource();
    String drugName = similarDrug.getProperty(RDFS.label).getString();
    String price = similarDrug.getProperty(priceProperty).getString();
    System.out.println("Drug: " + drugName + " - Price: " + price);
});
```

```
Resource drug = model.getResource("http://purl.org/net/hifm/data#986712");

Property similarTo = model.getProperty("http://purl.org/net/hifm/data#986712/hifm-ont:similarTo");

drug.listProperties(similarTo).forEachRemaining(statement -> {
    Resource similarDrug = statement.getObject().asResource();
    String drugName = similarDrug.getProperty(RDFS.label).getString();
    System.out.println("Similar drug: " + drugName);
});

Property priceProperty = model.getProperty("http://purl.org/net/hifm/data#987123/hifm-ont:refPriceWithVAT");

String drugPrice = drug.getProperty(priceProperty).getString();
System.out.println("Price of selected drug: " + drugPrice);

drug.listProperties(similarTo).forEachRemaining(statement -> {
    Resource similarDrug = statement.getObject().asResource();
    String drugName = similarDrug.getProperty(RDFS.label).getString();
    String price = similarDrug.getProperty(priceProperty).getString();
    System.out.println("Drug: " + drugName + " - Price: " + price);
});
```

Изработил:
Бојан Ристов (211151)