Домашно бр. 1

Jena RDF API

А) Прашања

1. **Како се изразува еден запис (еден факт) во RDF моделот?**
   - Секој факт во RDF моделот се изразува преку RDF тројка: (предмет, предикат, објект).

2. **Кои различни синтакси за RDF моделот постојат? Изразете го следниот факт во неколку различни RDF синтакси: „ВБС се предава на ФИНКИ“. Користете го префиксот @prefix finki: <http://finki.ukim.mk/resource#> за URI вредностите на ентитетите и релацијата.**
   - Постојат повеќе синтакси: Turtle, RDF/XML, RDFa, JSON-LD.

   Turtle: finki:vbs rdf:type finki:Predmet. finki:finki rdf:type finki:Fakultet. finki:vbs finki:ePredmetNa finki:finki

   RDF/XML:

```
<rdf:Description rdf:about="http://finki.ukim.mk/resource#vbs">
        <finki:ePredmetNa rdf:resource="http://finki.ukim.mk/resource#finki"/>
</rdf:Description>
```

   JSON-LD:

```
{
 "@context": {
  "finki": "http://finki.ukim.mk/resource#"
 },
 "@id": "finki:vbs",
 "@type": "finki:Predmet"
 "finki:ePredmetNa": {
  "@id": "finki:finki",
  "@type": "finki:Fakultet"
 }
}
```

3. **За што се користи RDF Schema?**

   - За валидација на RDF синтаксите, поставување на ограничувања и избегнување на контрадикции

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.**

   finki:Predmet rdf:type rdfs:Class. finki:Fakultet rdf:type rdfs:Class. finki:ePredmetNa rdf:Property. finki:ePredmetNa rdfs:domain finki:Predmet. finki:ePredmetNa rdfs:range finki:Fakultet.

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.
   - Ја приклучив Jena како dependency со помош на **Maven**

```
<dependencies>

    <dependency>
        <groupId>org.apache.jena</groupId>
        <artifactId>apache-jena-libs</artifactId>
        <version>3.16.0</version>
        <type>pom</type>
    </dependency>

</dependencies>
```

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.

```java
public static void main(String[] args) {

    Model model = ModelFactory.createDefaultModel();

    Resource dejan = model.createResource( uri: "https://www.linkedin.com/in/dejan-slamkov/", FOAF.Person);
    Resource ttmk = model.createResource( uri: "https://github.com/SlamkovDejan/ttmk", FOAF.Project);
    Resource finkiWebpage = model.createResource( uri: "https://finki.ukim.mk/en", FOAF.Document);
    Resource venko = model.createResource( uri: "https://www.linkedin.com/in/venko-stojanov-3970751b4", FOAF.Person);

    dejan.addProperty(FOAF.name, o: "Dejan Slamkov", l: "en")
            .addProperty(FOAF.name, o: "Дејан Сламков", l: "mk")
            .addProperty(FOAF.gender, o: "male", l: "en")
            .addProperty(FOAF.gender, o: "Машко", l: "mk")
            .addProperty(FOAF.birthday, o: "11-07")
            .addProperty(FOAF.age, o: "22")
            .addProperty(FOAF.made, ttmk)
            .addProperty(FOAF.pastProject, ttmk)
            .addProperty(FOAF.schoolHomepage, finkiWebpage)
            .addProperty(FOAF.knows, venko);

    venko.addProperty(FOAF.maker, ttmk).addProperty(FOAF.knows, dejan);
    ttmk.addProperty(FOAF.maker, dejan).addProperty(FOAF.maker, venko);

    model.write(System.out, lang: "TTL");

}
```

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():”.

```java
System.out.println("\tPrinting with model.listStatements():");
StmtIterator statementsIter = model.listStatements();
while (statementsIter.hasNext()){
    Statement statement = statementsIter.nextStatement();

    String subject = statement.getSubject().toString();
    String predicate = statement.getPredicate().toString();

    RDFNode node = statement.getObject();
    String object = node instanceof Resource ? node.toString() : "\"" + node.toString() + "\"";

    System.out.printf("%s - %s - %s\n", subject, predicate, object);
}
```

7. Извршете ја програмата. Може да го извршите целиот проект или само класата во која го дефинирвте кодот до сега. Проверете дали излезот на конзола соодветствува со она што очекувате да се прикаже. Дали сите RDF тројки кои ги дефинирвте во кодот, ги гледате отпечатени?
   - Сите тројки се испечатени

```
    Printing with model.listStatements():
https://finki.ukim.mk/en - http://www.w3.org/1999/02/22-rdf-syntax-ns#type - http://xmlns.com/foaf/0.1/Document
https://www.linkedin.com/in/dejan-slamkov/ - http://xmlns.com/foaf/0.1/age - "22"
https://www.linkedin.com/in/dejan-slamkov/ - http://xmlns.com/foaf/0.1/birthday - "11-07"
https://www.linkedin.com/in/dejan-slamkov/ - http://xmlns.com/foaf/0.1/knows - https://www.linkedin.com/in/venko-stojanov-3970751b4
https://www.linkedin.com/in/dejan-slamkov/ - http://xmlns.com/foaf/0.1/gender - "male@en"
https://www.linkedin.com/in/dejan-slamkov/ - http://xmlns.com/foaf/0.1/name - "Dejan Slamkov@en"
https://www.linkedin.com/in/dejan-slamkov/ - http://xmlns.com/foaf/0.1/gender - "машко@mk"
https://www.linkedin.com/in/dejan-slamkov/ - http://xmlns.com/foaf/0.1/schoolHomepage - https://finki.ukim.mk/en
https://www.linkedin.com/in/dejan-slamkov/ - http://xmlns.com/foaf/0.1/made - https://github.com/SlamkovDejan/ttmk
https://www.linkedin.com/in/dejan-slamkov/ - http://xmlns.com/foaf/0.1/pastProject - https://github.com/SlamkovDejan/ttmk
https://www.linkedin.com/in/dejan-slamkov/ - http://xmlns.com/foaf/0.1/name - "Дејан Сламков@mk"
https://www.linkedin.com/in/dejan-slamkov/ - http://www.w3.org/1999/02/22-rdf-syntax-ns#type - http://xmlns.com/foaf/0.1/Person
https://github.com/SlamkovDejan/ttmk - http://xmlns.com/foaf/0.1/maker - https://www.linkedin.com/in/venko-stojanov-3970751b4
https://github.com/SlamkovDejan/ttmk - http://xmlns.com/foaf/0.1/maker - https://www.linkedin.com/in/dejan-slamkov/
https://github.com/SlamkovDejan/ttmk - http://www.w3.org/1999/02/22-rdf-syntax-ns#type - http://xmlns.com/foaf/0.1/Project
https://www.linkedin.com/in/venko-stojanov-3970751b4 - http://xmlns.com/foaf/0.1/knows - https://www.linkedin.com/in/dejan-slamkov/
https://www.linkedin.com/in/venko-stojanov-3970751b4 - http://xmlns.com/foaf/0.1/maker - https://github.com/SlamkovDejan/ttmk
https://www.linkedin.com/in/venko-stojanov-3970751b4 - http://www.w3.org/1999/02/22-rdf-syntax-ns#type - http://xmlns.com/foaf/0.1/Person
```

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.

```java
String[] langs = {"RDF/XML", "RDF/XML-ABBREV", "N-TRIPLE", "Turtle"};
for(String lang : langs){
    System.out.printf("\tPrinting with model.write(), in %s\n", lang);
    model.write(System.out, lang);
    System.out.println();
}
```

9. Извршете ја програмата. Може да го извршите целиот проект или само класата во која го дефинирате 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?
   - Најкратка содржина има “Turtle” форматот. Најлесно се чита “N-TRIPLE” форматот според мене, кој исто така најлесно би можел да го процесирам.

```
    Printing with model.write(), in RDF/XML
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:j.0="http://xmlns.com/foaf/0.1/">
  <j.0:Project rdf:about="https://github.com/SlamkovDejan/ttmk">
    <j.0:maker>
      <j.0:Person rdf:about="https://www.linkedin.com/in/venko-stojanov-3970751b4">
        <j.0:knows>
          <j.0:Person rdf:about="https://www.linkedin.com/in/dejan-slamkov/">
            <j.0:age>22</j.0:age>
            <j.0:birthday>11-07</j.0:birthday>
            <j.0:knows rdf:resource="https://www.linkedin.com/in/venko-stojanov-3970751b4"/>
            <j.0:gender xml:lang="en">male</j.0:gender>
            <j.0:name xml:lang="en">Dejan Slamkov</j.0:name>
            <j.0:gender xml:lang="mk">машко</j.0:gender>
            <j.0:schoolHomepage>
              <j.0:Document rdf:about="https://finki.ukim.mk/en"/>
            </j.0:schoolHomepage>
            <j.0:made rdf:resource="https://github.com/SlamkovDejan/ttmk"/>
            <j.0:pastProject rdf:resource="https://github.com/SlamkovDejan/ttmk"/>
            <j.0:name xml:lang="mk">Дејан Сламков</j.0:name>
          </j.0:Person>
        </j.0:knows>
        <j.0:maker rdf:resource="https://github.com/SlamkovDejan/ttmk"/>
      </j.0:Person>
    </j.0:maker>
    <j.0:maker rdf:resource="https://www.linkedin.com/in/dejan-slamkov/"/>
  </j.0:Project>
</rdf:RDF>
```

```
    Printing with model.write(), in RDF/XML-ABBREV
<rdf:RDF
    xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
    xmlns:j.0="http://xmlns.com/foaf/0.1/">
  <j.0:Project rdf:about="https://github.com/SlamkovDejan/ttmk">
    <j.0:maker>
      <j.0:Person rdf:about="https://www.linkedin.com/in/venko-stojanov-3970751b4">
        <j.0:knows>
          <j.0:Person rdf:about="https://www.linkedin.com/in/dejan-slamkov/">
            <j.0:age>22</j.0:age>
            <j.0:birthday>11-07</j.0:birthday>
            <j.0:knows rdf:resource="https://www.linkedin.com/in/venko-stojanov-3970751b4"/>
            <j.0:gender xml:lang="en">male</j.0:gender>
            <j.0:name xml:lang="en">Dejan Slamkov</j.0:name>
            <j.0:gender xml:lang="mk">машко</j.0:gender>
            <j.0:schoolHomepage>
              <j.0:Document rdf:about="https://finki.ukim.mk/en"/>
            </j.0:schoolHomepage>
            <j.0:made rdf:resource="https://github.com/SlamkovDejan/ttmk"/>
            <j.0:pastProject rdf:resource="https://github.com/SlamkovDejan/ttmk"/>
            <j.0:name xml:lang="mk">Дејан Сламков</j.0:name>
          </j.0:Person>
        </j.0:knows>
        <j.0:maker rdf:resource="https://github.com/SlamkovDejan/ttmk"/>
      </j.0:Person>
    </j.0:maker>
    <j.0:maker rdf:resource="https://www.linkedin.com/in/dejan-slamkov/"/>
  </j.0:Project>
</rdf:RDF>
```

```
    Printing with model.write(), in N-TRIPLE
<https://finki.ukim.mk/en> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Document> .
<https://www.linkedin.com/in/dejan-slamkov/> <http://xmlns.com/foaf/0.1/age> "22" .
<https://www.linkedin.com/in/dejan-slamkov/> <http://xmlns.com/foaf/0.1/birthday> "11-07" .
<https://www.linkedin.com/in/dejan-slamkov/> <http://xmlns.com/foaf/0.1/knows> <https://www.linkedin.com/in/venko-stojanov-3970751b4> .
<https://www.linkedin.com/in/dejan-slamkov/> <http://xmlns.com/foaf/0.1/gender> "male"@en .
<https://www.linkedin.com/in/dejan-slamkov/> <http://xmlns.com/foaf/0.1/name> "Dejan Slamkov"@en .
<https://www.linkedin.com/in/dejan-slamkov/> <http://xmlns.com/foaf/0.1/gender> "машко"@mk .
<https://www.linkedin.com/in/dejan-slamkov/> <http://xmlns.com/foaf/0.1/schoolHomepage> <https://finki.ukim.mk/en> .
<https://www.linkedin.com/in/dejan-slamkov/> <http://xmlns.com/foaf/0.1/made> <https://github.com/SlamkovDejan/ttmk> .
<https://www.linkedin.com/in/dejan-slamkov/> <http://xmlns.com/foaf/0.1/pastProject> <https://github.com/SlamkovDejan/ttmk> .
<https://www.linkedin.com/in/dejan-slamkov/> <http://xmlns.com/foaf/0.1/name> "Дејан Сламков"@mk .
<https://www.linkedin.com/in/dejan-slamkov/> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .
<https://github.com/SlamkovDejan/ttmk> <http://xmlns.com/foaf/0.1/maker> <https://www.linkedin.com/in/venko-stojanov-3970751b4> .
<https://github.com/SlamkovDejan/ttmk> <http://xmlns.com/foaf/0.1/maker> <https://www.linkedin.com/in/dejan-slamkov/> .
<https://github.com/SlamkovDejan/ttmk> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Project> .
<https://www.linkedin.com/in/venko-stojanov-3970751b4> <http://xmlns.com/foaf/0.1/knows> <https://www.linkedin.com/in/dejan-slamkov/> .
<https://www.linkedin.com/in/venko-stojanov-3970751b4> <http://xmlns.com/foaf/0.1/maker> <https://github.com/SlamkovDejan/ttmk> .
<https://www.linkedin.com/in/venko-stojanov-3970751b4> <http://www.w3.org/1999/02/22-rdf-syntax-ns#type> <http://xmlns.com/foaf/0.1/Person> .
```

```
    Printing with model.write(), in Turtle
<https://finki.ukim.mk/en>
        a       <http://xmlns.com/foaf/0.1/Document> .

<https://www.linkedin.com/in/dejan-slamkov/>
        a       <http://xmlns.com/foaf/0.1/Person> ;
        <http://xmlns.com/foaf/0.1/age>
                "22" ;
        <http://xmlns.com/foaf/0.1/birthday>
                "11-07" ;
        <http://xmlns.com/foaf/0.1/gender>
                "male"@en , "машко"@mk ;
        <http://xmlns.com/foaf/0.1/knows>
                <https://www.linkedin.com/in/venko-stojanov-3970751b4> ;
        <http://xmlns.com/foaf/0.1/made>
                <https://github.com/SlamkovDejan/ttmk> ;
        <http://xmlns.com/foaf/0.1/name>
                "Dejan Slamkov"@en , "Дејан Сламков"@mk ;
        <http://xmlns.com/foaf/0.1/pastProject>
                <https://github.com/SlamkovDejan/ttmk> ;
        <http://xmlns.com/foaf/0.1/schoolHomepage>
                <https://finki.ukim.mk/en> .

<https://github.com/SlamkovDejan/ttmk>
        a       <http://xmlns.com/foaf/0.1/Project> ;
        <http://xmlns.com/foaf/0.1/maker>
                <https://www.linkedin.com/in/venko-stojanov-3970751b4> , <https://www.linkedin.com/in/dejan-slamkov/> .

<https://www.linkedin.com/in/venko-stojanov-3970751b4>
        a       <http://xmlns.com/foaf/0.1/Person> ;
        <http://xmlns.com/foaf/0.1/knows>
                <https://www.linkedin.com/in/dejan-slamkov/> ;
        <http://xmlns.com/foaf/0.1/maker>
                <https://github.com/SlamkovDejan/ttmk> .
```

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.

```
Model model = ModelFactory.createDefaultModel();
model.read( url: "./src/main/java/lab1/data/model.xml",  lang: "RDF/XML");
model.write(System.out,  lang: "TTL");
```

14. Извршете ја програмата. Може да го извршите целиот проект или само класата во која го дефинирате кодот до сега. Проверете дали излезот на конзола соодветствува со она што очекувате да се прикаже.
    - Додадени се префикси

```
@prefix rdf:   <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix j.0:   <http://xmlns.com/foaf/0.1/> .

<https://www.linkedin.com/in/dejan-slamkov/>
        a                   j.0:Person ;
        j.0:age             "22" ;
        j.0:birthday        "11-07" ;
        j.0:gender          "male"@en , "машко"@mk ;
        j.0:knows           <https://www.linkedin.com/in/venko-stojanov-3970751b4> ;
        j.0:made            <https://github.com/SlamkovDejan/ttmk> ;
        j.0:name            "Dejan Slamkov"@en , "Дејан Сламков"@mk ;
        j.0:pastProject     <https://github.com/SlamkovDejan/ttmk> ;
        j.0:schoolHomepage  <https://finki.ukim.mk/en> .

<https://www.linkedin.com/in/venko-stojanov-3970751b4>
        a          j.0:Person ;
        j.0:knows  <https://www.linkedin.com/in/dejan-slamkov/> ;
        j.0:maker  <https://github.com/SlamkovDejan/ttmk> .

<https://github.com/SlamkovDejan/ttmk>
        a          j.0:Project ;
        j.0:maker  <https://www.linkedin.com/in/dejan-slamkov/> , <https://www.linkedin.com/in/venko-stojanov-3970751b4> .

<https://finki.ukim.mk/en>
        a       j.0:Document .
```

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?

```
System.out.println("\tProperties of the entity 'https://www.linkedin.com/in/dejan-slamkov/':");
Resource dejan = model.getResource( uri: "https://www.linkedin.com/in/dejan-slamkov/");
StmtIterator iterator = dejan.listProperties();
while (iterator.hasNext()){
    Statement curr = iterator.nextStatement();

    String propertyId = curr.getPredicate().toString();
    RDFNode node = curr.getObject();
    String objectId = node instanceof Resource ? node.toString() : "\"" + node.toString() + "\"";

    System.out.printf("%s : %s\n", propertyId, objectId);
}
```

17. Извршете ја програмата. Може да го извршите целиот проект или само класата во која го дефинирате кодот до сега. Проверете дали излезот на конзола соодветствува со она што очекувате да се прикаже.

```
    Properties of the entity 'https://www.linkedin.com/in/dejan-slamkov/':
http://xmlns.com/foaf/0.1/age : "22"
http://xmlns.com/foaf/0.1/birthday : "11-07"
http://xmlns.com/foaf/0.1/knows : https://www.linkedin.com/in/venko-stojanov-3970751b4
http://xmlns.com/foaf/0.1/gender : "male@en"
http://xmlns.com/foaf/0.1/name : "Dejan Slamkov@en"
http://xmlns.com/foaf/0.1/gender : "машко@mk"
http://xmlns.com/foaf/0.1/schoolHomepage : https://finki.ukim.mk/en
http://xmlns.com/foaf/0.1/made : https://github.com/SlamkovDejan/ttmk
http://xmlns.com/foaf/0.1/pastProject : https://github.com/SlamkovDejan/ttmk
http://xmlns.com/foaf/0.1/name : "Дејан Сламков@mk"
http://www.w3.org/1999/02/22-rdf-syntax-ns#type : http://xmlns.com/foaf/0.1/Person
```

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.

```
Model model = ModelFactory.createDefaultModel();
model.read( url: "./src/main/java/lab1/data/hifm-dataset.ttl",  lang: "TTL");
```

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 научената материја 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.

```
Property typeProp = model.getProperty("http://www.w3.org/1999/02/22-rdf-syntax-ns#type");
RDFNode drugObject = model.getRDFNode(NodeFactory.createURI("http://wifo5-04.informatik.uni-mannheim.de/drugbank/resource/drugbank/drugs"));
Property nameProp = model.getProperty("http://wifo5-04.informatik.uni-mannheim.de/drugbank/resource/drugbank/brandName");

System.out.println("\tEvery drug in the system:");
ResIterator drugSubjects = model.listResourcesWithProperty(typeProp, drugObject);
ArrayList<String> drugNames = new ArrayList<>();
while (drugSubjects.hasNext()){
    Resource drugSubject = drugSubjects.next();
    String drugName = drugSubject.getProperty(nameProp).getString();
    drugNames.add(drugName);
}
drugNames.stream().sorted().forEach(System.out::println);
```

23. Одберете еден лек од графот (моделот) и за него излистајте ги сите релации и вредности. Click here

```
Resource oneDrug = model.getResource( uri: "http://purl.org/net/hifm/data#83496");
String oneDrugName = oneDrug.getProperty(nameProp).getString();

System.out.printf("\tDetails about %s (%s):\n", oneDrugName, oneDrug.getURI());
Task2.printResourcePropertiesWithValues(oneDrug);
```

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.

```
System.out.printf("\tDrugs similar to %s (%s):\n", oneDrugName, oneDrug.getURI());
Property similarToProp = model.getProperty("http://purl.org/net/hifm/ontology#similarTo");
StmtIterator similarDrugs = oneDrug.listProperties(similarToProp);
while (similarDrugs.hasNext()){
    String similarDrugURI = similarDrugs.nextStatement().getObject().toString();
    String similarDrugName = model.getResource(similarDrugURI).getProperty(nameProp).getString();
    System.out.println(similarDrugName);
}
```

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.

```
Property pricProp = model.getProperty("http://purl.org/net/hifm/ontology#refPriceWithVAT");
double price = oneDrug.getProperty(pricProp).getDouble();
System.out.printf("\tSimilar drugs to %s (%s) - %.2fMKD:\n", oneDrugName, oneDrug.getURI(), price);
similarDrugs = oneDrug.listProperties(similarToProp);
while (similarDrugs.hasNext()){
    String similarDrugURI = similarDrugs.nextStatement().getObject().toString();
    Resource similarDrug = model.getResource(similarDrugURI);
    String similarDrugName = similarDrug.getProperty(nameProp).getString();
    double similarDrugPrice = similarDrug.getProperty(pricProp).getDouble();
    System.out.printf("%s (%.2fMKD)\n", similarDrugName, similarDrugPrice);
}
```

26. Извршете ја програмата. Може да го извршите целиот проект или само класата во која го дефинирате кодот до сега. Проверете дали излезот на конзола соодветствува со она што очекувате да се прикаже.

```
    Every drug in the system:
ABAKTAL табл.10 x 400mg
ACIKLOVIR ALKALOID Крема 50mg/g (5g)
ACIKLOVIR ALKALOID Маст за очи 30mg/g(5g)
ACIKLOVIR ALKALOID табл. 30 x 200mg
ACIKLOVIR Крема 50mg/g (5g)
ACIKLOVIR табл. 25 x 200mg
ACIKLOVIR табл. 25 x 200mg
ACIPAN гастрорезистентни таблети 14X40mg
ACIPAN гастрорезистентни таблети 28X20mg
ACTONEL филм обл.табл.4x35mg
ADIABEN табл. 90 x 0,5mg
ADIABEN табл. 90 x 1mg
ADIABEN табл. 90 x 2mg
```

```
    Details about OMEZOL капс. 14 x 20mg (http://purl.org/net/hifm/data#83496):
http://www.w3.org/2000/01/rdf-schema#label : "Omeprazole"
http://purl.org/net/hifm/ontology#packaging : "14^^http://www.w3.org/2001/XMLSchema#integer"
http://wifo5-04.informatik.uni-mannheim.de/drugbank/resource/drugbank/brandName : "OMEZOL капс. 14 x 20mg"
http://purl.org/net/hifm/ontology#similarTo : http://purl.org/net/hifm/data#986712
http://purl.org/net/hifm/ontology#id : "83496^^http://www.w3.org/2001/XMLSchema#integer"
http://purl.org/net/hifm/ontology#similarTo : http://purl.org/net/hifm/data#975397
http://www.w3.org/1999/02/22-rdf-syntax-ns#type : http://purl.org/net/hifm/ontology#Drug
http://www.w3.org/2000/01/rdf-schema#seeAlso : http://wifo5-04.informatik.uni-mannheim.de/drugbank/resource/drugs/DB00338
http://purl.org/net/hifm/ontology#refPriceNoVAT : "45.71^^http://www.w3.org/2001/XMLSchema#decimal"
http://purl.org/net/hifm/ontology#similarTo : http://purl.org/net/hifm/data#993662
http://purl.org/net/hifm/ontology#refPriceWithVAT : "48.0^^http://www.w3.org/2001/XMLSchema#decimal"
http://www.w3.org/1999/02/22-rdf-syntax-ns#type : http://wifo5-04.informatik.uni-mannheim.de/drugbank/resource/drugbank/drugs
http://wifo5-04.informatik.uni-mannheim.de/drugbank/resource/drugbank/genericName : "Omeprazole"
http://purl.org/net/hifm/ontology#strength : "20mg"
http://purl.org/net/hifm/ontology#similarTo : http://purl.org/net/hifm/data#975354
http://www.w3.org/2000/01/rdf-schema#seeAlso : http://wifo5-04.informatik.uni-mannheim.de/drugbank/resource/drugs/DB00736
http://purl.org/net/hifm/ontology#manufacturer : "ALKALOID"
http://purl.org/net/hifm/ontology#similarTo : http://purl.org/net/hifm/data#950076
http://purl.org/net/hifm/ontology#dosageForm : "Капсули"
http://wifo5-04.informatik.uni-mannheim.de/drugbank/resource/drugbank/atcCode : "A02BC01"
http://purl.org/net/hifm/ontology#similarTo : http://purl.org/net/hifm/data#82481
http://purl.org/net/hifm/ontology#similarTo : http://purl.org/net/hifm/data#75949
```

```
    Drugs similar to OMEZOL капс. 14 x 20mg (http://purl.org/net/hifm/data#83496):
ULKOBOS капс. 14 x 40mg
ULKOBOS капс. 14 x 20mg
OMEPRAZID капс. 14 x 20mg
OMEPRAZOL капс. 14 x 20mg
OMEPRAZOL капс. 14 x 20mg
OMEPROL капс. 15 x 20mg
ULTOP капс. 14 x 20mg
    Similar drugs to OMEZOL капс. 14 x 20mg (http://purl.org/net/hifm/data#83496) - 48,00MKD:
ULKOBOS капс. 14 x 40mg (97,00MKD)
ULKOBOS капс. 14 x 20mg (48,00MKD)
OMEPRAZID капс. 14 x 20mg (48,00MKD)
OMEPRAZOL капс. 14 x 20mg (48,00MKD)
OMEPRAZOL капс. 14 x 20mg (48,00MKD)
OMEPROL капс. 15 x 20mg (52,00MKD)
ULTOP капс. 14 x 20mg (48,00MKD)
```

Source code

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.