

Vizsga maximális pontszám: 51 Megfelelt szint: 40%

Teszt kérdések (max. 11 pont)

Útmutató: Karikázza be a megfelelő választ, minden kérdésnél egy válasz jelölhető meg.
A helyes válasz kérdésenként 1 pontot ér.

1. Melyik szemantikus technológia leírásához szükséges definiálnunk szintaktikát és szemantikát is?
A URI
B OWL
C RDFS
D Mindegyikhez az előzőek közül

2. Az adattárházak tervezésénél, építésénél támaszkodhatunk-e ontológiákra?
A Igen, mert idővariáns adattárolási megközelítést alkalmazunk.
B Igen, mert témaorientált adattárolási megközelítést alkalmazunk.
C Nem, mert az ontológiák tervezésénél nyílt világ feltételezésre támaszkodunk.
D Nem, mert az adattárolás nem lehet illékony.

3. A local-as-view virtuális információ integráció megközelítés előnyös tulajdonsága, hogy
A Hierarchikus, többrétegű nézet struktúra megvalósítható.
B Nagy megbízhatóságú adatelérést nyújt.
C Források tartalmának leírását alkalmazhatjuk.
D Gyors lekérdezési lehetőségeket biztosít.

4. Melyik adatra nem vonatkozik a Dublin Core szabvány?
A Szerző
B Születési dátum
C Cím
D Létrehozás időpontja

5. Melyik elemet nem tudjuk közvetlenül definiálni egy ontológiában?
A Egy fogalom tárgyterületét
B Egy fogalom ellentettjét
C Egy fogalomhoz tartozó példányok számosságát
D Egy fogalom valószínűségét

6. Melyik állítás nem igaz a BDD-re?
A Előnye, hogy segít a korai hibamegtalálásban és az esetleges félreértések tisztázásában.
B A specifikáció leírásához Given/When/Assert szerkezetű nyelvet használ.
C A specifikációs példákban szereplő részlépésekhez automatizáló kódrészleteket lehet társítani.
D Egyik megvalósítása a Cucumber nevű eszköz.

7. Melyik állítás igaz statikus analízis eszközökkel kapcsolatban?
- A A statikus analízis eszköz által megtalált hibákat mindig ki kell javítani.
 - B A statikus analízis eszközök csak olyan hibákat tudnak megtalálni, amik előre definiált hibamintákra illeszkednek.
 - C A statikus analízis eszköz során figyelni kell a hamis pozitív és hamis negatív eredményekre.
 - D Statikus analízis eszközök szkript nyelvekhez nem érhetőek el.
8. Az a teszthelyettesítő, ami képes a hozzá intézett hívásokat is ellenőrizni, az a
- A stub
 - B fake
 - C mock
 - D spy
9. Melyik állítás nem igaz az ekvivalencia partícionálás módszerével kapcsolatban?
- A Egy-egy ekvivalencia osztályból egy-egy tesztadatot választunk ki.
 - B A konkrét ekvivalencia partíciók algoritmusok segítségével meghatározhatók.
 - C Kezdsnek érdemes az érvényes és érvénytelen bemenetek osztályait szétválasztani.
 - D Az érvényes ekvivalencia partíciókat kombináljuk, hogy minél kevesebb tesztesettel lefedhetők legyenek.
10. Melyik az a tesztelési szint, ahol nagy hangsúlyt kapnak a nem-funkcionális jellemzők ellenőrzése?
- A modul
 - B rendszer
 - C integrációs
 - D elfogadási
11. Melyik állítás igaz a különböző telepítési stratégiákra?
- A Continuous Deployment esetén az alkalmazásból mindig csak egy verzió van telepítve.
 - B Kék-zöld telepítés használata esetén nehéz visszaállni egy hiba esetén a működő verzióra.
 - C Kanári telepítés esetén az új verziót csak a felhasználók egy része látja.
 - D Az új verzió telepítésére a legjobb módszer az éles környezet felülírása az új verzióval.

Kidolgozandó feladatok (max. 40 pont)

1. Vizsgálja meg a következő RDF dokumentumot:

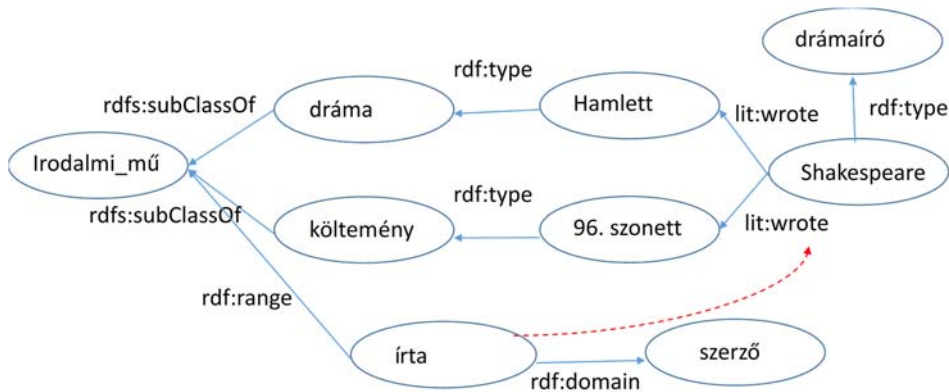
```
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:rdfs="http://www.w3.org/2000/01/rdf-schema#"
  xmlns:lit="http://literature.org/#"
  xml:base="http://literature.org/">
  <rdf:Description rdf:about=","#Hamlet">
    <rdf:type rdf:resource=","#dráma"/>
  </rdf:Description>
  <rdf:Description rdf:about=","#96._szonett">
    <rdf:type rdf:resource=","#költemény"/>
  </rdf:Description>
  <rdf:Description rdf:about=","#írta">
    <rdf:type rdf:resource=","#rdf:Property"/>
    <rdf:domain rdf:resource=","#szerző"/>
    <rdf:range rdf:resource=","#irodalmi_mű"/>
  </rdf:Description>
  <rdfs:Class rdf:about=","#költemény">
    <rdfs:subClassOf rdf:resource=","#irodalmi_mű"/>
  </rdfs:Class>
  <rdfs:Class rdf:about=","#dráma">
    <rdfs:subClassOf rdf:resource=","#irodalmi_mű"/>
  </rdfs:Class>
  <lit:poet rdf:about=","#Shakespeare">
    <lit:wrote rdf:resource=","#96. szonett"/>
    <lit:wrote rdf:resource=","#Hamlet"/>
    <rdf:type rdf:resource=","#drámairó"/>
  </lit:poet>
</rdf:RDF>
```

- a) Milyen információt ír le a fenti RDF(S) dokumentum? Fogalmazzon meg magyarul egy lehetséges interpretációt! (2 pont)

Megoldás:

A „Hamlet” egy dráma, a „96. szonett” egy költemény. Az írásnak, mint alkotásnak az alanya az író, tárgya pedig az irodalmi mű. A vers és a dráma irodalmi művek. Shakespeare egy drámairó típusú költő, ő írta a Hamlet-et és a 96. szonettet.

b) Mutassa be az RDF(S) dokumentum gráf reprezentációját (2 pont)



(Ha valaki az ábrán összekapcsolta vagy egy elemként vette fel a „lit:wrote” és az „írtá” elemeket az helyes, ebben az esetben egy „rdf:Property” kapcsolat és egy üres csomópont felvételével jelölheti ezt a gráfban.)

c) Adjon meg egy SPARQL lekérdezést, amely kigyűjti az RDF(S) dokumentumból az irodalmi műveket! (2 pont)

Megoldás:

```

PREFIX <rdf:RDF xmlns:rdf=„http://www.w3.org/1999/02/22-rdf-syntax-ns#"
        xmlns:rdfs=„http://www.w3.org/2000/01/rdf-schema#"
        xmlns:lit=„http://literature.org/#"
        xml:base=„http://literature.org/">
SELECT ?content {
  ?content_type rdfs:subClassOf „#irodalmi_mű” .
  ?content rdf:type ?content_type}
  
```

d) A következők közül melyik kijelentések írhatók le RDF/S-ben? Válaszait indokolja! (3 pont)

- A költők és a drámaírók szerzők.
- A költők verseket írnak, a drámaírók drámákat.
- Költők nem írnak RDF dokumentumokat.

Megoldás:

- A költők és a drámaírók szerzők.
Ez az állítás két RDF(S) állításban leírható úgy, hogy mind a költők, mind a drámaírók alosztályai a szerzőknek.

- A költők verseket írnak, a drámaírók drámákat.

Ez az állítás közvetlenül nem fogalmazható meg RDF(S)-ben, mert az írás állítmánynak (tulajdonságnak) csak egy értékkeszlete és egy tárgyterülete lehet. Ha definiálunk olyan tárgyterület halmazt, ami valamilyen módon tartalmazza a verset és a drámát is, akkor ezzel a kiegészítéssel az állítást már le lehet írni RDF(S)-ben.

(Aki a megoldásában ez utóbbi megjegyzésre hivatkozva arra a következtetésre jut, hogy az állítás megfogalmazható RDF(S)-ben, annak a választát elfogadjuk.)

- Költők nem írnak RDF dokumentumokat.

Ez az állítás nem reprezentálható RDF(S)-ben, mert az RDF(S) nem tartalmaz negációt.

2. feladat

(a) Nevezzen meg 3-3 előnyös tulajdonságát a lokális és a globális megközelítésnek a nézet alapú virtuális információ integrációs technikák esetében! (3 pont)

Megoldás:

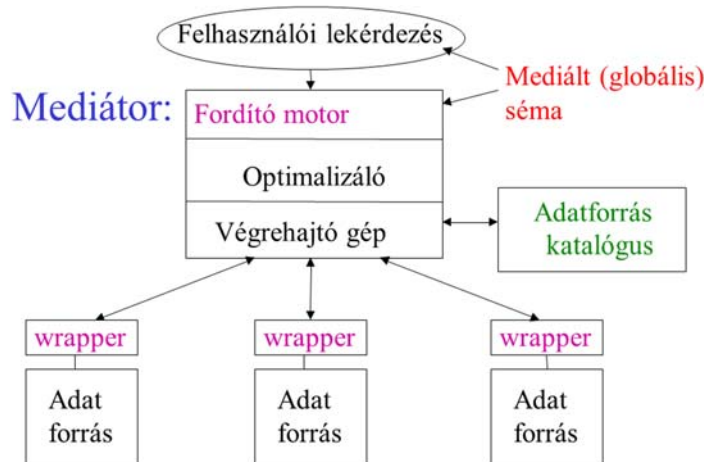
Globális nézet alapú megközelítés előnyei:

- egyszerű a lekérdezések lefordítása
- moduláris, hierarchikus leképezések megvalósíthatóak
- kis számú forrás esetében egyszerű a definiálása

Lokális nézet alapú megközelítés előnyei:

- rugalmas nézet átalakítás változós források esetén
- forrás tulajdonságokat könnyű figyelembe venni
- a saját koncepcionális modellünkben fogalmazhatunk

(b) Mutasson be egy integrátor/mediátor rendszer architektúrát, amely alkalmas webes információforrások adatainak integrációját támogatni ! Ismertesse a rendszer fontosabb elemeinek szerepét a lekérdezések átalakításában! (3 pont)



A fordító motor felelős a lekérdezéseknek a források elemeire vonatkozó újraírásáért. Az átalakított lekérdezések hatékony futtatása céljából az optimalizáló előállítja a lekérdezési tervet. A végrehajtó gép végzi a lekérdezések ütemezett futtatását az egyes források felé. A wrapper-ek biztosítják, hogy a források saját leíró nyelve/technológiája és a mediátor technológiája között kapcsolatot teremtsen, szintaktikai átírást biztosítson.

Feladatok

1. Specifikáció ellenőrzése

Hírolvasó alkalmazásunkba tervezünk egy új ajánló funkciót, amihez a következő kezdeti követelményleírást kaptuk. Az ajánló funkció a felhasználó által beállított hírforrások és a felhasználó olvasási szokásai alapján ajánl további releváns híreket és hírforrásokat. A rendszerünk szerveroldali komponense a hírforrásokat tartalmuk alapján csoportosítja, és ez alapján tud ajánlani további forrást. Azokat a hírforrásokat, amiket a felhasználó gyakrabban olvas, nagyobb súllyal veszi figyelembe az ajánlások során. Az ajánlások az elolvasott hírek alatt jelennek meg, ezzel is biztatva a felhasználót a tovább olvasásra.

Ellenőrizzük a kapott specifikációt. Milyen kérdéseink és észrevételeink lennének? [5 pont]

MO:

1. A meglévő hírek elolvasása nélkül nem is lehet hozzájutni az ajánlásokhoz?
2. Működik-e az ajánló rendszer, ha nem olvas híreket a felhasználó, de vannak felvett hírforrásai?
3. Keveredik a magas szintű specifikáció és a UI terv. Hasonló módon a funkciók architektúráján belüli szétosztása (mi szervert és mi kliens oldal) lehet, hogy most még korai.
4. Hogyan működik pontosan a súlyozás? Olvasásnál relatív vagy abszolút gyakoriságokat akarunk mérni (más-más lehet az egyes hírforrások frekvenciája)?
5. A csoportosítás mi alapján megy? Lehetnének képek is a hírekben vagy csak szöveg? Csak a szöveget veszi figyelembe? Mit teszünk, ha különböző nyelvű, de azonos témájú hírforrások is vannak?
6. ...

2. Forráskód átvizsgálása

Adott a következő forráskód részlet.

```
String trim(String input,int n)
{
    if (n>0) {
        input.substring(n , input.length());
    }
    return input;
}
```

a) Soroljon fel legalább két problémát a kód stílusával kapcsolatban! [4 pont]

b) Soroljon fel legalább két olyan problémát, amely potenciális hibalehetőséget rejt magában! [4 pont]

MO:

a)

- inkonzisztens zárójelezés,
- inkonzisztens szóközök az operátorok között,
- változóelnevezés sem túl informatív

b)

- immutable string nem változik,
- input lehet null,
- „n” lehet nagyobb, mint a sztring hossza

3. Specifikáció-alapú tesztelés

Egy pályázati rendszere a pályázatok költségvetését ellenőrző funkciót kell tesztelnünk. A pályázatokhoz költség tételeket rendelhetünk. Minden pályázathoz legalább egy tételt meg kell adni. Egy tétel személyi kifizetés vagy beszerzés lehet. A személyi kifizetésre lehet akár 100%-os támogatást igényelni, a beszerzésre legfeljebb 75%-ost. A pályázaton igényelhető támogatási összeg 1 millió Ft-ban van limitálva.

- a) Milyen tesztervezési technikát lenne célszerű alkalmazni a tesztesetek megtervezése során?
Válaszát indokolja! [2 pont]
- b) Milyen teszteseteket választana ki a funkció minél teljesebb ellenőrzése érdekében? [4 pont]

MO:

- a) Ekvivalencia partíciók használata az egyes paraméterekre, hogy a fontosabb eseteket lefedjük.

Utána ezt lehet bővíteni a határértékek vizsgálatával, mert több határértéket is definiál a leírás, és ezeknek a kezelése okozhat hibát.

- b) Ekvivalencia partíciók meghatározása:

7. tételek száma: 0 (?), 1, több

8. tétel típusa: személyi, beszerzés

9. támogatás mértéke: negatív (?), 0%, $0% < x < 75%$, 75%, 100%, 100%-nál nagyobb (?)

10. támogatási tételek összege: negatív (?), 0, $0 < x < 1000000$, 1000000, 1000000-nál nagyobb

Ezek alapján az partíciókat kombinálva lehet konkrét értékeket is kiválasztani:

Teszt	Típus	Támogatás mértéke	Költség tétel összege (támogatás nélkül)	Eredmény
1	-	-	-	HIBA
2	személyi	-10%	1000	HIBA
3	beszerzés	110%	10000	HIBA
4	beszerzés	80%	5000	HIBA
5	személyi	100%	-1500	HIBA
6	beszerzés	50%	5000000	HIBA
7	beszerzés	50%	1500000	OK (a teljes összeg >1M Ft, de a támogatási nem)
8	személyi	70%	500000	OK
	beszerzés	70%	200000	
9	személyi	100%	500000	OK (határértékek)
	beszerzés	75%	200000	
10	személyi	50%	1000000	OK (támogatási összeg határa)
	személyi	80%	625000	
11	személyi	0%	500000	?? (megengedünk 0 Ft támogatást?)
12	beszerzés	0%	200000	OK? (támogatás nem 0 Ft, de van 0%-os tétel)
	személyi	20%	3000	
...				

Még lehetne további tesztek is felvinni, de első körben ennyi lehet elég is. Ha minden helyesen működik, akkor lehet további kombinációkat vagy speciális eseteket nézni.

4. Struktúra-alapú tesztelés

Adott a következő forráskód részlet.


```

int pow(int n, int k) {
    if (n < 0 || k < 0) {
        return -1;
    }

    int p = 1;
    for (int i = 0; i < k; i++) {
        p *= n;
    }
    return p;
}

```

a) Hány százalékos döntés lefedettséget ér el az alábbi tesztkészlet? [3 pont]

Teszteset	n	k
T1	-1	5
T2	3	0

b) Adjon meg egy tesztkészletet, amely 100%-os utasítás lefedettséget garantál! [3 pont]

MO:

a)

n	k	$n < 0 \ \ k < 0$	$i < k$
-1	5	igaz	nem értékelődik ki
3	0	hamis	hamis

→ $3/4 = 75\%$

b)

Két teszt elég hozzá (egy olyan, ahol $n < 0$ vagy $k < 0$ és egy olyan, ahol $n \geq 0$ és $k > 0$), pl.

n	k
-1	5
3	1

3.