

**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 szabvány foglalkozik dokumentumok tulajdonságainak megfogalmazásával?
  - a. RDFS
  - b. FOAF
  - c. Dublin Core
  - d. DBPedia
  
2. Két RDFS osztályra megfogalmazható a következő tulajdonságok:
  - a. Diszjunktság
  - b. Ekvivalencia
  - c. Számosság
  - d. Egyik sem a fentiek közül
  
3. Egy inverz szabályok módszerére épülő mediátor rendszer, melyik komponense implementálja az algoritmust?
  - a. Lekérdezés fordító.
  - b. Lekérdezés optimalizáló.
  - c. Lekérdezés végrehajtó.
  - d. Wrapper.
  
4. Milyen szerepe van egy üres csomópontnak egy RDF gráfban?
  - a. Több kijelentés tárgyát foghatjuk össze vele.
  - b. Egy tulajdonság leírásához több specifikusabb tulajdonságot kapcsolunk.
  - c. Változók helyét jelölhetjük.
  - d. Példánnyal nem rendelkező osztályt jelöl.
  
5. Melyik állítást nem tudjuk megfogalmazni OWL2 nyelven?
  - a. Minden diák szeret tanulni
  - b. Van olyan diák, aki szeret szórakozni.
  - c. Lehet olyan diák aki szeret dolgozni.
  - d. Minden diák ember.
  
6. Melyik állítás nem igaz a Linked Data technológiára?
  - a. Minden alkalmazott URI érvényes URL is egyben
  - b. Tetszőleges adatbázisok között kapcsolatot teremt
  - c. Dokumentumok adatait kapcsoljuk össze
  - d. Minden adatnak URI-val azonosítottnak kell lennie
  
7. BDD, TDD és ATDD: melyik állítás igaz?
  - a. A BDD és a TDD ugyanaz, csak különböző csoportok más nevet használnak rá.
  - b. A BDD épít a TDD elveire és előnyeire, és a két technika jól használható együtt.
  - c. Az ATDD az Advanced TDD rövidítése, ez a technika egyfajta továbbfejlesztése.
  - d. A technikákat a következő sorrendben kell alkalmazni a fejlesztés során: ATDD, BDD, TDD.

8. Melyik állítás igaz kódolási irányelvekkel kapcsolatban?
  - a. A MISRA egy olyan ajánlás, amit vasútipari, C nyelvű szoftverek fejlesztésekor használható.
  - b. A kódolási irányelvek olyan szabályhalmazok, amik ajánlásokat fogalmaznak meg a forráskóddal kapcsolatban.
  - c. A kódolási irányelvek minden esetben kötelezően betartandó szabályokat tartalmaznak.
  - d. Java nyelvű program fejlesztésekor a Google kódolási irányelveit érdemes használni.
9. Melyik nem úgynevezett test smell egységtesztelés esetén?
  - a. A teszt kód sok feltételes logikát tartalmaz, hogy kevesebb tesztesetet kelljen írni.
  - b. A tesztelt egység függőségeit izoláljuk, hogy azok hibái ne befolyásolják a vizsgált működést.
  - c. A tesztek sok assertion hívást tartalmaznak, hogy lehetőleg mindent leellenőrizzenek.
  - d. A teszt kód minden egyes teszt előtt újra inicializálja az alkalmazás adatbázisát.
10. Használati esetekből (use case) milyen tesztek nem szokás származtatni
  - a. rendszer
  - b. egység (unit)
  - c. elfogadási
  - d. integrációs
11. A tesztelési piramis javaslata szerint hogyan érdemes automatizálni a tesztekét?
  - a. Nem érdemes integrációs tesztekkel bajlódni, a rendszerteszték úgyis ellenőrzik a komponensek integrációját is.
  - b. Készítsünk sok GUI tesztet, mert azzal sok funkciót le lehet fedni.
  - c. A unit tesztekkel kellően ellenőrzött kód tud biztos alapot nyújtani a további tesztelésnek.
  - d. A manuális tesztek teljesen el lehet hagyni, az automatizálás megtalálja minden hibát.

**Kidolgozandó feladatok (max. 40 pont)**

1. A gyakorlaton az alábbi lekérdezést alkalmaztuk művészeti alkotásokkal kapcsolatos adatok elemzéséhez.

a.) Adjon meg egy olyan RDF adatmodellt leíró gráfot, amelyre a lekérdezés eredményt ad!  
(3 pont)

b.) Adjon meg egy lehetséges természetes nyelvű interpretációt az a.) feladatra adott válaszában felvázolt RDF adatmodellhez!  
(2pont)

```
PREFIX ecrm: http://erlangen-crm.org/current/  
SELECT ?actor ?creation ?object {  
  ?actor a ecrm:E39_Actor .  
  ?creation ecrm:P11_had_participant ?actor ;  
    a ecrm:E65_Creation .  
  ?object ecrm:P12i_was_present_at ?creation ;  
    a ecrm:E18_Physical_Thing .  
}
```

2. Adatmodellezés

a) Ismertesse a nyílt világ feltételezés és a zár világ feltételezés lényegét  
(3 pont)

b) Adjon egy példát egy konkrét adatmodellre és egy kapcsolódó lekérdezésre, ahol a nyílt világ feltételezés és a zár világ feltételezés különböző eredményt ad.  
(2 pont)

3. Tekintsük a következő relációs sémát az S1 és S2 forrásokra, amelyben budapesti színházak előadásairól gyűjtünk adatokat. Az adatbázis a színházakra vonatkozóan tárolja a színház nevét, az előadott darabok címét, és a színház címét, illetve a játszott színművek címét és szerzőjét:
- Színház(Színháznev, Színdarab\_cím, Utca\_Házzám),  
Színművek(Színdarab\_cím, Szerző),

A kapcsolódó mediált sémánkban a színdarabokkal és a darabok előadásával kapcsolatban lehet a következő hasonló információkat lekérdezni:

Színdarab(Színdarab\_cím, Szerző, Színháznev),  
Helyszín(Színháznev, Város, Utca\_Házzám).

(a) Adjon meg egy local-as-view leképezést a források és a mediált séma között! (3 pont)

(b) Adjon meg egy lekérdezést a mediált sémánkon azoknak a színházak nevének és címének (Utca\_Házzám) legyűjtéséhez, amelyek játszanak Shakespeare művet! (2 pont)

(A feladat megoldásához használhat datalog, SQL vagy SPARQL szintaxist is.)

(c) Írja át a (b) pontban megfogalmazott lekérdezést a forrásokon futtatható lekérdezéssé a veder algoritmus alkalmazásával!

Adja meg a vedreket az ezekhez tartozó nézetek felsorolásával! (2 pont)

Adja meg az átalakított lekérdezést! (3 pont)

---

#### 4. Forráskód átvizsgálása

Adott a következő forráskód részlet.

```
public Token nextToken(Reader r) throws IOException {
    int c = r.read();
    while (c != -1){
        switch (c) {
            case ' ': case '\t':
            case '\n':
                c = r.read();
            case '(':
                return Token.LPAREN;

            case ')':
                return new Token(c, TokenType.RPAREN);
            default:
                return new Token(c, TokenType.CHAR);
        }
    }
    return Token.EOF;
}
```

- a) Soroljon fel legalább két problémát a kód formázásával és stílusával kapcsolatban! [2 pont]
- b) Soroljon fel legalább két olyan problémát, amely potenciális hibalehetőséget rejt magában! [2 pont]

#### 5. Specifikáció-alapú tesztelés

Egy banki alkalmazáshoz fejlesztünk egy olyan modult, ami kedvezményes kamatozású betéteket javasol az ügyfeleknek. Ha valakinek legalább 5 millió Ft a számlájának az egyenlege, akkor az aktuális betéti kamat + 1% prémiumot, ha legalább 10 millió Ft az egyenlege, akkor +2%-ot javasol. Ha teljesül az is, hogy az elmúlt 3 hónapban mindig legalább 300e Ft jövedelem érkezett a számlájára, akkor még +1% prémiumot kap.

- a) Ellenőrizzük a kapott specifikációt. Milyen kérdéseink és észrevételeink lennének? [2 pont]
- b) Mik az egyes paraméterek ekvivalencia-osztályai és határértékei? [2 pont]
- c) Táblázatos formában adja meg, hogy milyen konkrét teszteseteket választana ki az alkalmazás minél teljesebb ellenőrzése érdekében, ha legfeljebb 6 tesztesetet definiálhat! [4 pont]

## 6. Struktúra-alapú tesztelés

Adott a következő forráskód részlet.

```
int collatz(int n) {
    while (n > 1) {
        if (n % 2 == 0) {
            n /= 2;
        } else {
            n = 3 * n + 1;
        }
    }
    return n;
}
```

- Rajzolja fel a függvény vezérlési folyam gráfját (CFG)! [4 pont]
- Adjon meg pontosan egy tesztesetet, amely 100%-os döntés lefedettséget garantál! Állítását indokolja a lefutás megadásával! [4 pont]