AF-SM-04\_05.2020.G

|  |  |
| --- | --- |
| **კურსის სახელწოდება:** | **დაპროგრამება JVM პლატფორმაზე (I)** |
| **ლექტორი:** | **ნათია მესტვირიშვილი** |
| **სტუდენტი:** | **მარიამ მეკარიშვილი** |
| **თარიღი:** | **16/01/2022** |

**ვარიანტი II**

**ინსტრუქციები**

**პროექტის სახელი:**

პროექტის შექმნისთვის გამოიყენეთ თქვენი სახელი და გვარი, კერძოდ, თუ ჩემი სახელი და გვარია ნათია მესტვირიშვილი, პროექტის სახელი უნდა ჯდებოდეს შემდეგ ფორმატში:

JVM\_Final\_Natia\_Mestvirishvili

**მონაცემთა ბაზის სახელი:**

იგივე რაც პროექტის სახელი

**თეორიული კითხვარის word ფაილის სახელი:**

იგივე რაც პროექტის სახელი

**გამოცდის ჩაბარების ფორმატი:**

Zip/rar არქივი იგივე სახელწოდებით რაც აქვს პროექტს. არქივი უნდა შეიცავდეს:

1. პრაქტიკული დავალების ჯავა პროექტს
2. თეორიული კითხვარის word ფაილს
3. მონაცემთა ბაზის ექსპორტ ფაილს sql გაფართოებით

**მონაცემთა ბაზის ექსპორტის ინსტრუქცია:**

MySQL Workbench-ის ნავიგაციის მენიუში Server -> Data export -> ვნიშნავთ სასურველ სქემას -> გადავდივართ Export progress თაბზე -> Start export -> ფაილი ინახება მითითებულ ლოკაციაზე

**საჭირო Dependencies:**

*Log4J Slf4j ვრაპერით:*

<dependency>

<groupId>org.apache.logging.log4j</groupId>

<artifactId>log4j-slf4j-impl</artifactId>

<version>2.9.0</version>

</dependency>

*Mysql ქონექთორი:*

<dependency>

<groupId>mysql</groupId>

<artifactId>mysql-connector-java</artifactId>

<version>8.0.24</version>

</dependency>

**Log4j2 ფრეიმვორკის კონფიგურაცია**

<?xml version="1.0" encoding="UTF-8"?>

<Configuration status="INFO">

<Appenders>

<Console name="ConsoleAppender" target="SYSTEM\_OUT">

<PatternLayout pattern="%d{HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n" />

</Console>

<File name="FileAppender" fileName="application-${date:yyyyMMdd}.log" immediateFlush="false" append="true">

<PatternLayout pattern="%d{yyy-MM-dd HH:mm:ss.SSS} [%t] %-5level %logger{36} - %msg%n"/>

</File>

</Appenders>

<Loggers>

<Root level="debug">

<AppenderRef ref="ConsoleAppender" />

<AppenderRef ref="FileAppender"/>

</Root>

</Loggers>

</Configuration>

1. ჩამოთვალეთ პოლიმორფიზმის ორი ტიპი (1 ქულა)

პოლიმორფიზმის ორი ტიპი არსებობს: Compile-Time და Runtime

1. დაასახელეთ JVM არქიტექტურის კომპონენტი, რომელიც გამოუყენებელ ობიექტებს ავტომატურად ასუფთავებს (1 ქულა)

Garbage Collector

1. დაასახელეთ მინიმუმ ერთი განსხვავება Set-სა და List-ს შორის (1 ქულა)

List-ისგან განსხვავებით, Set მოიცავს უნიკალურ ელემენტებს

1. მოკლედ აღწერეთ logstash-ის დანიშნულება ELK სთექში (1 ქულა)

Logstash გვჭირდება სისტემის ლოგების წასაკითხად და ElasticSearch-ში გადასაგზავნად

1. რისთვისაა საჭირო ჯავაში Thread კლასი ან Runnable ინტერფეისი (1 ქულა)

საშუალებას გვაძლევს რამდენიმე ტასკი პარალელურად შევასრულოთ

1. მოკლედ აღწერეთ, რა არის ანბოქსინგი (1 ქულა)

wrapper-ის ტიპის ობიექტის შესაბამის პრიმიტიულ მნიშვნელობად გადაყვანას ეწოდება "unboxing"

1. ჩამოთვლილთაგან რომელი კლასი ახდნეს List ინტერფეისის იმპლემენტაციას (1 ქულა)
2. Vector
3. PriorityQueue
4. TreeSet
5. TreeMap
6. დაასახელეთ მინიმუმ ერთი განსხვავება HashSet-სა და TreeSet-ს შორის (1 ქულა)

HashSet უფრო სწრაფია

1. რა დანიშნულებას ასრულებს ჯავაში Stream API? (1 ქულა)
2. მუშაობს მონაცემთა ბაზასთან
3. გვეხმარება კოლექციების გაფილტვრასა და სხვა დამხმარე ოპერაციების შესრულებაში
4. გვეხმარება პროექტის დაბილდვასა და ბიბლიოთეკების მართვაში
5. გვეხმარება Desktop აპლიკაციების აგებაში
6. რა დანიშნულებას ასრულებს ტესტირებაში Mockito ფრეიმვორკი? (1 ქულა)
7. გვეხმარება ტესტირებისას ყალბი (Dummy) ობიექტების შექმნაში
8. გვეხმარება პროექტის დაბილდვასა და ბიბლიოთეკების მართვაში
9. გვეხმარება unit ტესტირებაში

**დავალება 1 (10 ქულა)**

1. შექმენით Maven პროექტი და დაამატეთ dependencies: MySQL Connector და Log4J Slf4j ვრაპერით (2 ქულა)
2. შექმენით მონაცემთა ბაზა ინსტრუქციებში მოცემული სახელით (1 ქულა)
3. შექმენით ცხრილი Animal შემდეგი column-ებით:

id (Primary Key, Not Null, Auto Increment), ტიპი int

name (Not Null), ტიპი varchar

weight (Not Null), ტიპი int

1. შექმენით კლასი, რომელიც დაუკავშირდება თქვენს მონაცემთა ბაზას და Student ცხრილში ჩაწერს 1 ახალ row-ს (3 ქულა)

**დავალება 2 (10 ქულა)**

1. შექმენით ენუმერაცია სახელად Animal, სადაც გექნებათ სამი წევრი TIGER, LION, RABBIT(2 ქულა)
2. Pets ენუმერაციაში თითოეულ წევრს უნდა ჰქონდეს String ტიპის მნიშნვლობა. TIGER -ის შემთხვევაში “Tiger”, LION -ის შემთხვევაში “Lion”, ხოლო RABBIT -ის შემთხვევაში “Rabbit”. ენუმერაციაში უნდა გაითვალისწინოთ რომ სტრიქონული მნიშნვლობის ჩასაწერად ენუმერაციას შესაბამისი String property და კონსტრუქტორი უნდა ჰქონდეს. დაა-override-ეთ toString() მეთოდი რომ დააბრუნოს ენუმერაციის სტრიქონული მნიშვნელობა (3 ქულა)
3. შექმენით კლასი EnumExample. ამ კლასში შექმენით მეთოდი (სახელი თავად მოუფიქრეთ), რომელსაც პარამეტრად მიაწოდებთ Animal ტიპის ენუმერაციას (2 ქულა)
4. მესამე სთეფში შექმნილი მეთოდის პარამეტრი მოაქციეთ switch statement-ში და თუ იგი უდრის Animal.TIGER-ს, დაპრინტეთ “This is a Tiger”, თუ უდრის Animal.LION-ს, მაშინ დაპრინტეთ “This is a Lion”, თუ უდრის Pet.RABBIT -ს, მაშინ დაპრინტეთ “This is a Rabbit” (3 ქულა)

**დავალება 3 (10 ქულა)**

1. გამართეთ slf4j ლოგირების ვრაპერი log4j ფრეიმვორკით ისე, რომ მინიმალური ლოგირების დონე იყოს debug. (2 ქულა)
2. შექმენით HashMap სახელად Programmers, რომელსაც ექნება String ტიპის key და int ტიპის value (key წარმოადგენს სახელსა და გვარს, value წარმოადგენს მათი ნამუშევარი წლების რაოდენობას). (2 ქულა)
3. ჩაამატეთ რამდენი წევრი Programmers კოლექციაში. (1 ქულა)
4. Stream API-სა და Lambda expression-ის დახმარებით გაფილტრეთ Programmers კოლექცია და ახალ HashMap-ში ჩაწერეთ მხოლოდ ის წევრები, რომელთა value აღემატება 3-ს. (3 ქულა)
5. დავალება 5-ის კოდი მოაქციეთ try ბლოკში და catch ბლოკში ლოგირება მოახდინეთ კონსოლსა და ლოგირების ფაილში (2 ქულა)