Java exercises

overload

class Person{  
   private String firstname;  
   private String lastname;  
   private int birthmonth;  
   private int birthday;  
   private int birthyear;  
   private String ssn;  
   public Person(String aFirstname, String aLastname, int aBirthmonth, int aBirthday, int aBirthyear, String aSsn) {  
       super();  
       firstname = aFirstname;  
       lastname = aLastname;  
       birthmonth = aBirthmonth;  
       birthday = aBirthday;  
       birthyear = aBirthyear;  
       ssn = aSsn;  
   }  
   public String getFirstname() {  
       return firstname;  
   }  
   public String getLastname() {  
       return lastname;  
   }  
   public void setFirstname(String aFirstname) {  
       firstname = aFirstname;  
   }  
   public void setLastname(String aLastname) {  
       lastname = aLastname;  
   }  
   public String getBirthday() {  
       return birthmonth+"/"+birthday+"/"+birthyear;  
   }  
   public boolean verifySSN (String s) {  
       return s.equals(ssn);  
   }  
}  
public class Main {  
   public static void main(String[] args) {  
       Person p = new Person("Uday", "Kumar", 3, 22, 2000, "1234");  
       System.out.println(p.getBirthday());  
                
   }  
}

Method

**package** MethodPractice;

**public** **class** Practice3 {

**public** **static** **void** main (String [] args) {

**double** a = *info*(9.6,3.6);

System.***out***.println(a);

}

**public** **static** **double** info(**double** a, **double** b ) {

**double** netice =a-b;

**return** netice;

}

\*

**package** MethodPractice;

**public** **class** Practice4 {

**public** **static** **void** main (String [] args) {

*Kitab*(99, 11);

}

**public** **static** **void** Kitab (**int** b, **int** c ) {

**if** (b<c) {

System.***out***.println("riyaziyyat");

}

**else** {

System.***out***.println( "ana dili");

}

}

}

**METHOD RETURN**

**MESELE 1.**

**public** **static** **void** main ( String [] args) {

Scanner scan = **new** Scanner (System.***in***);

**int** number = *calculate*(7);

System.***out***.println(number);

}

**public** **static** **int** calculate (**int** practice) {

**return** practice + practice;

}

}

**CAAVAB 14. Cunki 7+7=14**

**MESELE 2.**

**public** **static** **void** main(String[] args) {

Scanner scan = **new** Scanner (System.***in***);

System.***out***.println("Enter student information");

String name = scan.next();

String lastname = scan.next();

String studentName = *findEmail*(name, lastname);

System.***out***.println(studentName);

}

**public** **static** String findEmail (String name, String lastname) {

**return** "welcome to the college";

}

}

**CAVAB**

Enter student information

sarah

mir

welcome to the college

**Scanner**

En boyuk reqemi tapin.

Scanner in = new Scanner(System.in);

Int a,b,c

System.out.println(“A ededini daxil edin”);

A=in.nextInt();

B=in.next.Int;

System.out.println(“B ededini daxil edin”);

C=in.nextInt;

If (a<b && a>c) {

System.out.println(“A ededi boyukdur”);

}

Else if (b>a && b>c) {

System.out.println(“b ededi boyukdur”);

}

Else if (c >a && c>v) {

System.out.println(“ ca ededi boyukdur”)

}

NESTED LOOP

1-den 10-a kimi her reqemi besh defe yaz.

**public** **static** **void** main(String[] args) {

**for** (**int** a = 1; a <= 10; a++) {

**for** (**int** b = 1; b <= 5; b++) {

System.***out***.println(a);

}

// System.out.println(" ");

Cavab

1

1

1

1

1

2

2

2

2

2

3

3

3

3

3

4

4

4

4

4

5

5

5

5

5

6

6

6

6

6

7

7

7

7

7

8

8

8

8

8

9

9

9

9

9

10

10

10

10

10

**Continue Loop**

25den sonra davam elesin eger

**for** (**int** n = 20; n <= 30; n++) {

**if** (n <= 25)

**continue**;

System.***out***.println("Otaglarin nomresi " + n);

Cavab

Otaglarin nomresi 26

Otaglarin nomresi 27

Otaglarin nomresi 28

Otaglarin nomresi 29

Otaglarin nomresi 30

**Break loop**

1. 7ye gelib catanda qirilsin arxasin yazmasin.

**for** (**int** a= 0; a <10; a++) {

**if** (a== 7)

**break**;

System.***out***.println("Reqemleri sayin " + a);

}

Cavab

Reqemleri sayin 0

Reqemleri sayin 1

Reqemleri sayin 2

Reqemleri sayin 3

Reqemleri sayin 4

Reqemleri sayin 5

Reqemleri sayin 6

@createExpense  @hw  
Scenario Outline: Create Expenses - Positive Scenario  
    Given User creates request data "<name>" , "<amount>", "<expenseDateTime>","<expenseNameID>" for OtherExpense API  
    And User submits POST request to "CreateExpense\_URL" api  
    And User validates if the statusCode is 200  
    Then User validate if value of name in response is "<responseName>"     Examples:  
     | name        | expenseNameID | expenseDateTime     | amount | responseName |  
     | Pizza  | 1             | 09/01/2020 01:36:29 | 555.5  | Electricity  |  
     | Pizza       | 2             | 08/30/2020 01:36:29 | 666.6  | Rent         |  
     | Pizza       | 3             | 07/28/2020 01:36:29 | 777.7  | Gas          |  
     | Pizza       | 4             | 09/14/2020 01:36:29 | 888.8  | Pizza        |

String actualValue=JsonPath.read(response.asString(), "$.result.name");  
 System.out.println(actualValue);  
 assertEquals(nameValue, actualValue);

${project\_loc:mealB}

<build>

<plugins>

<plugin>

<groupId>org.apache.maven.plugins</groupId>

<artifactId>maven-surefire-plugin</artifactId>

<version>3.0.0-M5</version>

<configuration>

<parallel>methods</parallel>

<threadCount>1</threadCount>

<perCoreThreadCount>false</perCoreThreadCount>

<testFailureIgnore>true</testFailureIgnore>

<includes>

<include>\*\*/\*Runner.java</include>

</includes>

</configuration>

</plugin>

<plugin>

<groupId>net.masterthought</groupId>

<artifactId>maven-cucumber-reporting</artifactId>

<version>4.8.0</version>

<executions>

<execution>

<id>execution</id>

<phase>verify</phase>

<goals>

<goal>generate</goal>

</goals>

<configuration>

<projectName>cucumber-jvm-example</projectName>

<!-- output directory for the generated report -->

<outputDirectory>${project.build.directory}</outputDirectory>

<!-- optional, defaults to outputDirectory if not specified -->

<inputDirectory>${project.build.directory}</inputDirectory>

<jsonFiles>

<!-- supports wildcard or name pattern -->

<param>\*\*/cucumber.json</param>

</jsonFiles>

<classificationFiles>

<!-- supports wildcard or name pattern -->

<param>sample.properties</param>

<param>other.properties</param>

</classificationFiles>

<!--<parallelTesting>false</parallelTesting>-->

</configuration>

</execution>

</executions>

</plugin>

</plugins>

</build>

[INFO] BUILD SUCCESS

[INFO] ------------------------------------------------------------------------

[INFO] Total time: 10.098 s

[INFO] Finished at: 2022-03-29T20:19:20-04:00

[INFO] ------------------------------------------------------------------------