

JAVA 11

3. forduló



A kategória támogatója: IBM

RENDELKEZÉSRE ÁLLÓ IDŐ:

10:00

Ismertető a feladathoz



Fontos információk

Ezután a forduló után automatikusan jár a [kitartóknak szóló garantált ajándékunk](#), érdemes kitöltened a feladatlapot! :)

Ha kifutsz az adott feladatlap kitöltésére rendelkezésre álló időből, a felület **automatikusan megpróbálja beküldeni** az addig megadott válaszokat.

A kérdésekre **mindig van helyes válasz**, olyan kérdés viszont nincs, amelyre az összes válasz helyes!

Egyéb információkat a [versenyszabályzatban](#) találsz!

Felhasznált idő: 02:06/10:00

Elért pontszám: 0/30

1. feladat 0/0 pont

Válaszd ki milyen időegységeket tárolhatunk egy Duration objektumban!

Válaszok

- ☐ Év
- ☐ Hónap
- ☐ Hét
- ☒ Nap
- ☒ Óra
- ☒ Perc
- ☒ Másodperc
- ☒ Milliszekundum
- ☒ Nanoszekundum

Magyarázat

Szervezői / hitelesítői információ

A kérdést az eredmények és az érkezett visszajelzések függvényében 0 pontosra állítottuk, ugyanis a válaszok azt mutatják, hogy sokféleképp lehetett értelmezni a kérdést.

Napot vagy annál kisebb időegységeket tudunk benne tárolni.

2. feladat 0/5 pont

Milyen dátumot reprezentál a date változó?

```
LocalDate date = LocalDate.of(2016, 1, 30);  
date.plusMonths(1);
```

Válasz

- ☐ 2016. Február 28.
- ☐ 2016. Február 29.
- ☐ 2016. Március 1.
- ☐ 2016. Január 30.

- ☐ 2016. Április 1.
- ☐ DateTimeException-t dob

Magyarázat

A `LocalDate` immutable típus, ezért a kiindulási dátumot tárolja a változó, függetlenül attól, hogy milyen metódusokat hívtunk rajta.

3. feladat 0/5 pont

Mely megoldások reprezentálják az alábbi dátumot?

2021. április 25.

Válaszok



```
LocalDate date = LocalDate.of(2021, Month.APRIL, 25);
```



```
LocalDate date = LocalDate.of(2021, Month.APRIL, 31).minusDays(6);
```



```
LocalDate date = LocalDate.of(2021, 4, 31).minusDays(36);
```



```
LocalDate date = LocalDate.of(2021, Month.MAY, 31).minusMonths(1).minusDays(5);
```



```
Calendar calendar = new GregorianCalendar(2021, 3, 25);  
Date date = calendar.getTime();
```



```
Calendar calendar = new GregorianCalendar(2021, 4, 25);  
Date date = calendar.getTime();
```



```
LocalDate date = new LocalDate(2021, Month.APRIL, 25);
```

Magyarázat

Április 31. érvénytelen dátum, ezért DateTimeException-t kapunk. A Calendar osztályok 0-tól sorszámozzák a hónapokat. A LocalDate konstruktora privát, így fordítási hibát kapunk ha megpróbáljuk meghívni.

4. feladat 0/5 pont

Mi kerül kiíratásra a konzolon, ha lefuttatjuk az alábbi kódrészletet?

```
LocalDateTime date = LocalDateTime.of(2015, 5, 10, 11, 22, 33);
Period period = Period.ofYears(1).ofMonths(2);
date = date.minus(period);
DateTimeFormatter formatter = DateTimeFormatter.ofLocalizedDate(FormatStyle.SHORT);
System.out.println(date.format(formatter));
```

Válasz

- ☐ 10/05/2014
- ☐ 10/03/2014
- ☒ 10/03/2015
- ☐ 10/05/2014, 11:22
- ☐ 10/03/2014, 11:22
- ☐ 10/03/2015, 11:22
- ☐ Nem fordul le a kód
- ☐ Futásidőben hibát dob

Magyarázat

Period metódusait nem lehet láncoltan hívni, amikor létrehozzuk, mert statikusak. Így csak a 2 hónapot tárolja el a period változóban. Ezen felül a formatter-rel megadjuk, hogy csak a dátumot jelenítsük meg.

5. feladat 0/5 pont

Válaszd ki az igaz állításokat!

2017-07-10T05:00 GMT-04:00

2017-07-10T10:00 GMT+02:00

Válaszok

- ☐ Az első kifejezés jelöli a korábbi időpontot
- ☒ A második kifejezés jelöli a korábbi időpontot
- ☐ A két kifejezés azonos időpontot jelöl
- ☒ A két időpont között 1 óra eltérés van
- ☐ A két időpont között 5 óra eltérés van
- ☐ A két időpont között 2 óra eltérés van

Magyarázat

GMT-ben átszámolva az első kifejezés 9:00-át jelöli a második pedig 8:00-át.

6. feladat 0/5 pont

Milyen dátumot reprezentál a date változó?

```
Calendar c = Calendar.getInstance();  
c.set(2016, 1, 30);  
c.add(Calendar.DATE, 2);  
Date date = c.getTime();
```

Válasz

- ☐ 2016. Január 30.
- ☐ 2016. Január 31.
- ☐ 2016. Február 1.
- ☐ 2016. Február 30.
- ☒ 2016. Március 3.
- ☐ 2016. Március 4.
- ☐ DateTimeException-t dob

Magyarázat

A régi Calendar API 0-tól kezdi a hónapok sorszámozását, ezért az 1. hó valójában február, és mivel 2016 szökőév így a kiindulási dátum március 1. A Calendar nem immutable, így ehhez hozzáadva 2 napot március 3-át kapjuk.

7. feladat 0/5 pont

Mi kerül kiíratásra a konzolon az alábbi kódrészlet lefutása után?

```
LocalDate date = LocalDate.of(2017, 9, 5);
LocalTime time = LocalTime.of(14, 30, 00);
ZoneId zone = ZoneId.of("Europe/Budapest");
ZonedDateTime zonedDateTime = ZonedDateTime.of(date, time, zone);
Instant instant = zonedDateTime.toInstant();
instant = instant.plus(1, ChronoUnit.WEEKS);
System.out.println(instant);
```

Válasz

- ☐ 2017-09-05T12:30:00Z
- ☐ 2017-09-12T12:30:00Z
- ☐ 2017-09-05T14:30:00Z
- ☐ 2017-09-12T14:30:00Z
- ☐ Nem fordul le az alábbi kódrészlet
- ☐ UnsupportedOperationException-t dob

Magyarázat

Habár egy Instant objektum jelöli a dátumot is, műveleteket végezni rajta csak nap és ettől kisebb ChronoUnit egységekkel lehet (pl. nap, óra, perc).