

Darbs ar laiku

1.uzdevums "TimeSpan"

- Izveidot programmu, kura demonstrē struktūras TimeSpan darbību.
- Main() funkcijā izveidot masīvu ar 10 TimeSpan tipa objektiem.
- Katra cikla solī programma pieprasa inicializē masīva TimeSpan tipa elementu;
- Izveidot funkciju AttēlotIpashibas(), kas saņem TimeSpan tipa vienu parametru un izvada tā vērtību uz ekrāna;
- Izvadīt visus masīva elementus uz ekrāna

Programmas izpildīšanas rezultāts

```
Inicializēt 1 objektu
TimeSpan : 1
```

```
Inicializēt 2 objektu
TimeSpan : 111222333444555
```

```
.....
```

Rezultāts

```
TimeSpan( 1 )
```

```
00:00:00.0000001
```

```
Days 0
```

TotalDays

```
1,15740740740741E-12
```

```
Hours 0
```

TotalHours

```
2,77777777777778E-11
```

```
Minutes 0
```

TotalMinutes

```
1,66666666666667E-09
```

```
Seconds 0
```

TotalSeconds

```
1E-07
```

```
Milliseconds 0
```

TotalMilliseconds

```
0,0001
```

Ticks

```
1
```

```
TimeSpan( 111222333444555 )
```

```
128.17:30:33.3444555
```

```
Days 128
```

TotalDays

```
128,729552597865
```

```
Hours 17
```

TotalHours

```
3089,50926234875
```

```
Minutes 30
```

TotalMinutes

```
185370,555740925
```

```
Seconds 33
```

TotalSeconds

```
11122233,3444555
```

```
Milliseconds 344
```

TotalMilliseconds

```
11122233344,4555
```

Ticks

```
111222333444555
```

2.uzdevums "Calendar"

- Izveidot programmu, kura demonstrē klases Calendar darbību.
- Main() funkcijā izveidot masīvu ar 10 DateTime objektiem.
- Funkcija Main() paziņot GregorianCalendar;
- Katra cikla solī programma pieprasa inicializēt masīva DateTime tipa elementu;
- Izveidot funkciju AttēlotKalendaru(), kas saņem DateTime tipa parametru un Calendar tipa parametru, un izvada tā vērtību uz ekrāna;
- Izvadīt visus masīva elementus uz ekrāna.

Programmas izpildīšanas rezultāts

Inicializēt 1 objektu

Gads: 2019

Mēnesis:11

Diena: 25

.....

Attēlot:

10. april, 2010 of the Gregorian calendar:

Era:	1
Year:	2019
Month:	10
DayOfYear:	329
DayOfMonth:	30
DayOfWeek:	Monday
Hour:	0
Minute:	0
Second:	0
Milliseconds:	0

.....

3.uzdevums

Apskatīt klases:

- HebrewCalendar
- JulianCalendar

No katras klases izvēlēties jūsuprāt svarīgus atribūtus un metodes un aprakstīt līdzīgi kā iepriekšējos uzdevumos. Izdomāt programmu, kura demonstrē izvēlētas klašu īpašības un metodes.

Struktūra DateTime

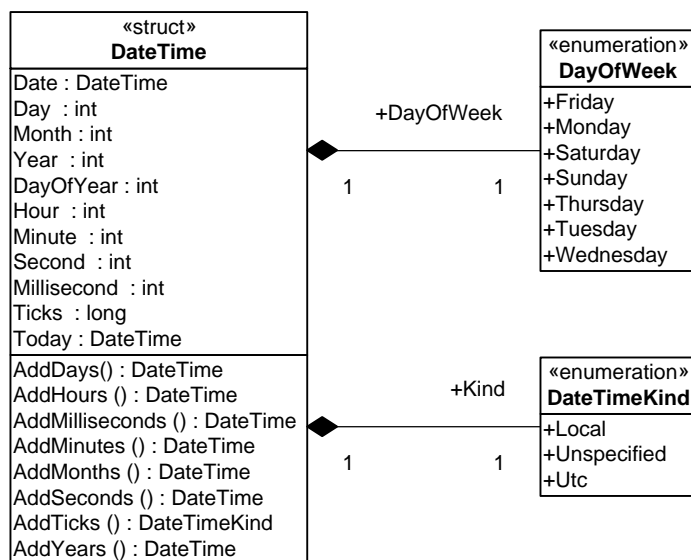
Attēlo laika vērtību, kas izteiktā, ka datums un laiks. DateTime struktūra attēlo datumu un laiku sākot no 12:00:00 pusnaktī, 1. Janvāra mūsu ēras līdz to 11:59:59 p.m 31. decembrim 9999.gadam mūsu ēras. Laika vērtības tika mērītas 100-nanosekunžu vienībās, kas tiek sauktas kā tiks.

Kā attēlots diagrammā, DateTime struktūra satur divus numerātorus:

- DayOfWeek, kas satur nedēļas dienu nosaukumus;
- Kind, kas satur laika veidu (lokāls, UTC, nav norādīts veids);

Īpašības

- Date – datums;
- Day – dienas numurs;
- Month – mēnesis;
- Year – gads;
- DayOfWeek – nedēļas diena;
- DayOfYear – dienas numurs gadā;
- Hour – stunda;
- Minute – minūte;
- Second – sekunde;
- Millisecond – milisekunde;
- Ticks – tiks;
- Today – statiskā īpašība, satur pašreizējo datumu;



Metodes

- AddDays(double value) – saņem dienu skaitu kā parametru un pievieno struktūras objektam;
- Addhours(double value) – saņem stundu skaitu kā parametru un pievieno struktūras objektam;
- AddMilliseconds(double value) – saņem milisekundes skaitu kā parametru un pievieno struktūras objektam;
- AddMinutes(double value) – saņem minūtes skaitu kā parametru un pievieno struktūras objektam;
- AddMonths(int value) – saņem mēnešu skaitu kā parametru un pievieno struktūras objektam;
- AddTicks(long value) – saņem tikšu skaitu kā parametru un pievieno struktūras objektam;
- AddSecond(double value) – saņem sekunžu skaitu kā parametru un pievieno struktūras objektam;
- AddYears(int value) – saņem gadu skaitu kā parametru un pievieno struktūras objektam;

Numerātors DayOfWeek

Satur nedēļas dienu nosaukumus.

Vērtības

- Fiday - Piektdiena
- Monday - Pirmdiena
- Saturday – Sestdiena
- Sunday – Svētdiena
- Thursday – Ceturtdiena
- Tuesday – otrdiena
- Wednesday – Trešdiena

Numerātors DateTimeKind

Norāda vai DateTime struktūras objekts attēlo lokālo laiku, Coordinated Universal Time (UTC), vai norāda, ka nav specificēts kā lokāls laiks vai UTC. Numerātorā DateTimeKind vērtības izmanto pāriešanā starp lokālo laiku un UTC, bet neizmanto laika salīdzināšanai vai aritmētiskām darbībām.

Vērtības

- Local – parāda, ka DateTime objekts ir specificēts attēlot lokālo laiku;
- UTC – parāda, ka DateTime objekts ir specificēts attēlot UTC;
- Unspecified – parāda, ka DateTime objekts nav specificēts attēlot lokālo laiku vai UTC.

Programmas piemērs

Programma demonstrē DateTime struktūras īpašību un metožu izmantošanas variantus.

```
public static void Main()
{
    DateTime dt = new DateTime();
    // paziņo DateTime objektu
    Console.WriteLine("Noklusējumā DateTime objekts ir
inicializēts ar : " +dt);
    Console.WriteLine("Šodienas datums: "+DateTime.Today);
    // statiskās īpašības Today izmantošana
    Console.WriteLine("Ievadiet dzimšanas datumu:");

    Console.Write("Diena : ");
    dt = dt.AddDays(int.Parse(Console.ReadLine())-1);
    //metodes AddDays() izmantošana
    Console.Write("Mēnesis: ");
    dt = dt.AddMonths(int.Parse(Console.ReadLine())-1);
    //metodes AddMonths () izmantošana
    Console.Write("Gads : ");
    dt=dt.AddYears(int.Parse(Console.ReadLine())-1);
    //metodes AddYears() izmantošana
```

```
Console.WriteLine("\nDzimšanas datums ir: "+dt.ToString());
Console.WriteLine("Jūsu dzimšanas dienas numurs ir: "+dt.DayOfYear); // īpašības DayOfYear izmantošana
Console.WriteLine("Jūsu dzimšanas diena ir: "+dt.DayOfWeek); // numerātorā DayOfWeek izmantošana
Console.WriteLine("Laika veids: "+dt.Kind.ToString());
// numerātorā Kind izmantošana
}
```

Programmas darbības rezultāts

Noklusējuma DateTime objekts ir inicializēts ar :
0001.01.01. 0:00:00
Šodienas datums: 2010.04.10. 0:00:00
Ievadiet dzimšanas datumu:
Diena : 1987
Mēnesis: 6
Gads : 6

Dzimšanas datums ir: 1987.06.06. 0:00:00
Jūsu dzimšanas dienas numurs ir: 157
Jūsu dzimšanas diena ir: Saturday
Laika veids: Unspecified

Programmas paskaidrojums

Programmā tika izveidots DateTime struktūras objekts. Uz ekrāna izveda objekta noklusējuma vērtību. Tāpēc, pievienojot objektam dienu, mēnešu, vai gadu skaitu ir jāatņem vienu (1) lai iegūtu precīzu datumu, nevis par vienu dienu, mēnesi, vai gadu vairāk. Objekts DateTime satur statistiskās īpašības.

```
Console.WriteLine("Šodienas datums: "+DateTime.Today);
```

Izveda uz ekrāna pašreizējo datumu.

```
dt = dt.AddDays(double.Parse(Console.ReadLine())-1);
```

Metode Console.ReadLine() atgriež simbolu rindu nolasītu no klaviatūras. Metode int.Parse() saņem kā parametru simbolu rindu un pārveido par skaitli int tipa. No iegūtā skaitļa atņem 1, lai precizētu datumu. Metode dt.AddDays() saņem kā parametra double tipa vērtību. Pievieno objektam DateTime iegūto vērtību ar metodes AddDays() palīdzību.

Struktūra TimeSpan

Struktūra reprezentē laika posmu. Vērtības diapazons ir mērīts tikos, un atrodas diapazonā starp Int64.MinValue un Int64.MaxValue. Tiks ir vismazākā laika vienība, kura ir vienāda 100 nanosekundēm. Gan tiks, gan TimeSpan vērtība var būt pozitīva vai negatīva.

TimeSpan vērtība var būt attēlota string formātā, kā "[-]d.hh:mm:ss.fff" kur "-" ir negatīvā zīmē, "d" attēlo dienas, "hh" – stundas, "mm" – minūtes, "ss" – sekundes, un "fff" – attēlo sekundes

«struct» TimeSpan
Days : int
Hour : int
Millisecond : int
Minute : int
Second : int
Ticks : long
TotalDays : double
TotalHours : double
TotalMilliseconds : double
TotalMinute : double
TotalSecond : double
Add() : TimeSpan
Compare() : TimeSpan
FromHours () : TimeSpan
FromMilliseconds () : TimeSpan
FromMinutes () : TimeSpan
FromSeconds () : TimeSpan
FromTicks () : TimeSpan

daļas. Piemēram, struktūra `TimeSpan` inicializēta, kā `1.0e+13` tikai reprezentē "11.13:46:40", kas ir 11 dienas, 13 stundas, 46 minūtes, un 40 sekundes.

Īpašības

- `Days` – dienas;
- `Hour` –stundas;
- `Millisecond` – milisekundes;
- `Minute` – minūtes;
- `Second` – sekundes;
- `Ticks` – tiks;
- `TotalDays` – dienu skaits veselās un daļās;
- `TotalHours` – stundu skaits veselās un daļās;
- `TotalMillisecond` – milisekunžu skaits veselās un daļās;
- `TotalMinute` – minūtes skaits veselās un daļās minūtēs;
- `TotalSecond` –sekunžu skaitu veselās un daļās sekundēs;

Metodes

Pārskaitītās metodes ir statiskās.

- `Add(TimeSpan ts)` – saņem `TimeSpan` objektu kā parametru pievieno `TimeSpan` objektam;
- `Compare(TimeSpan ts1, TimeSpan ts2)` – saņem divus `TimeSpan` objektus kā parametrus un salīdzina tos;
- `FormHours(double value)` – saņem stundu skaitu un atgriež `TimeSpan` objektu;
- `FromMilliseconds(double value)` – saņem milisekunžu skaitu un atgriež `TimeSpan` objektu;
- `FromMinutes(double value)` – saņem minūtes skaitu un atgriež `TimeSpan` objektu;
- `FormSeconds(double value)` – saņem sekunžu skaitu un atgriež `TimeSpan` objektu;
- `FromTicks(long value)` - saņem tiku skaitu atgriež `TimeSpan` objektu.

Programmas piemērs

Programma demonstrē `TimeSpan` struktūras īpašību un metožu izmantošanas variantus. Programmā izveidoti trīs `TimeSpan` objekti `t1`, `t2` un `t3`. Objekti `t1` un `t2` ir inicializēti ar konstruktoru, bet objekts `t3` ir inicializēts ar `t1` un `t2` starpību.

```
public static void Main()
{
    TimeSpan t1 = new TimeSpan(15, 30, 10); //objekta
    izveidošana un inicializēšana
    TimeSpan t2 = new TimeSpan(23, 45, 6);
    TimeSpan t3 = new TimeSpan();           // objekta
    izveidošana
    Console.WriteLine(t1.ToString());       //izvada
    vērtības uz ekrāna
    Console.WriteLine(t2.ToString());
    t3 = t2 - t1;
```

```

    Console.WriteLine("Laika t1 un t2 starpība
"+t3.ToString());
    t2 = TimeSpan.FromHours(3);           // metodes
FromHours() izmantošana
    t2 = TimeSpan.FromMinutes(15);        // metodes
FromMinutes() izmantošana
    t2 = TimeSpan.FromSeconds(1);         // metodes
FromSeconds() izmantošana
    t3 = t1 - t2;
    Console.WriteLine("Laika starpība pēc laika t2
palielināšanas " +t3.ToString());
}

```

Programmas izpildīšanas rezultāts

15:30:10

23:45:06

Laika t1 un t2 starpība 08:14:56

Laika starpība pēc laika t2 palielināšanas 15:30:09

Klase Calendar

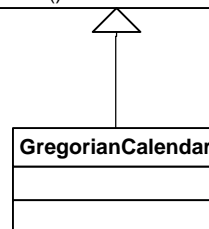
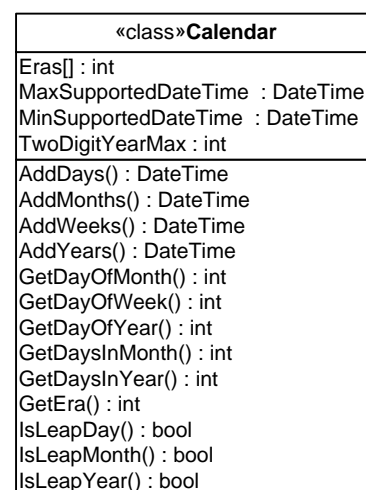
Klase reprezentē laiku dienās, mēnešos un gados. Klases ir abstraktā, ko nozīmē, ka nevar paziņot šīs klases objektu.

Īpašības

- Eras –ēru saraksts;
- MaxSupportedDateTime –pēdējais datums un laiks, kuru satur kalendārs;
- MinSupportedDateTime –pirmais datums un laiks, kuru satur kalendārs;
- TwoDigitYearMax –pēdējais no 100 gadiem gads, kuru var attēlot ar 2 cipariem;

Metodes

- AddDays(DateTime time, int days) – saņem DateTime objektu un dienu skaitu kā parametrus un pievieno dienu skaitu DateTime objektam;
- AddMonth(DateTime time, int months) – saņem DateTime objektu un mēnešu skaitu kā parametrus un pievieno mēnešu skaitu DateTime objektam;
- AddWeeks(DateTime time, int weeks) – saņem DateTime objektu un nedēļu skaitu kā parametrus un pievieno nedēļu skaitu DateTime objektam;
- AddYears(DateTime time, int years) – saņem DateTime objektu un gadu skaitu kā parametrus un pievieno gadu skaitu DateTime objektam;
- GetDayOfMonth(DateTime time) – saņem DateTime objektu kā parametru un atgriež noteiktā mēneša dienu;



Algoritmi un struktūras

- `GetDayOfWeek(DateTime time)` – saņem `DateTime` objektu kā parametru un atgriež noteiktās nedēļas dienu;
- `GetDayOfYear(DateTime time)` – saņem `DateTime` objektu kā parametru un atgriež noteiktā gada dienu;
- `GetDaysInMonth(DateTime time)` – saņem `DateTime` objektu kā parametru un atgriež dienu skaitu mēnesī;
- `GetEra(DateTime time)` – saņem `DateTime` objektu kā parametru un atgriež ēru;
- `IsLeapDay(int year, int month, int day)` – saņem gadu, mēnesi un pārbauda vai norādīta diena ir īsa;
- `IsLeapMonth(int year, int month)` – saņem gadu, mēnesi un pārbauda vai norādītais mēnesis ir īss;
- `IsLeapYear(int year)` – saņem gadu atgriež un pārbauda vai norādītais gads ir īss.

Programmas piemērs

Programma demonstrē klases `GregorianCalendar` izmantošanas variantus.

```
public static void DisplayValues(Calendar myCal, DateTime
myDT)
{
    Console.WriteLine("Era: {0}", myCal.GetEra(myDT));
    //metodes GetEra() izmantošana
    Console.WriteLine("Year: {0}", myCal.GetYear(myDT));
    //metodes GetYear() izmantošana
    Console.WriteLine("Month:{0}", myCal.GetMonth(myDT));
    //metodes GetMonth() izmantošana
    Console.WriteLine("DayOfYear: {0}",
myCal.GetDayOfYear(myDT)); //metodes GetDayOfYear()
izmantošana
    Console.WriteLine("DayOfMonth:{0}",
myCal.GetDayOfMonth(myDT)); //metodes GetDayOfMonth()
izmantošana
    Console.WriteLine("DayOfWeek: {0}",
myCal.GetDayOfWeek(myDT)); //metodes GetDayOfWeek()
izmantošana
    Console.WriteLine("Hour: {0}", myCal.GetHour(myDT));
    //metodes GetHour() izmantošana
    Console.WriteLine("Minute: {0}", myCal.GetMinute(myDT));
    //metodes GetMinute () izmantošana
    Console.WriteLine("Second: {0}", myCal.GetSecond(myDT));
    //metodes GetSecond() izmantošana
    Console.WriteLine("Milliseconds: {0}",
myCal.GetMilliseconds(myDT)); //metodes GetMilliseconds()
izmantošana
    Console.WriteLine();
}
public static void Main()
{
```


Algoritmi un struktūras

```
DateTime myDT = new DateTime(2005, 9, 11); //paziņo
DatiTime objektu un inicializē
GregorianCalendar myCal = new GregorianCalendar();
//paziņo GregorianCalendar objektu
DisplayValues(myCal, myDT); //funkcijas izsaukšana
Console.WriteLine("Pēc laika pievienošanas:\n");
myDT = myCal.AddHours(myDT, 15); //metodes AddHours()
izmantošana
myDT = myCal.AddMinutes( myDT, 14 ); //metodes
AddMinutes() izmantošana
myDT = myCal.AddSeconds( myDT, 22 ); //metodes
AddSeconds() izmantošana
myDT = myCal.AddMilliseconds(myDT, 7); //metodes
AddMilliseconds() izmantošana
DisplayValues(myCal, myDT); //funkcijas izsaukšana
}
```

Programmas izpildīšanas rezultāts

```
Era:          1
Year:         2005
Month:        9
DayOfYear:    254
DayOfMonth:   11
DayOfWeek:    Sunday
Hour:         0
Minute:       0
Second:       0
Milliseconds: 0
```

Pēc laika pievienošanas:

```
Era:          1
Year:         2005
Month:        9
DayOfYear:    254
DayOfMonth:   11
DayOfWeek:    Sunday
Hour:         15
Minute:       14
Second:       22
Milliseconds: 7
```