

RP3 2021/2022

2. zadaća

Rok predaje: 23. 1. 2022. u 23:59 ([Merlin](#))

U ovoj zadaći implementiramo aplikaciju za izradu i pregled linijskog ili raspršenog (po želji) grafa. U zadaći se ne boduje efikasnost (složenost) rješenja. Svi rubni slučajevi koji nisu pokriveni tekстом zadaće ostavljeni su vama na izbor. Prije implementacije pogledajte snimku jedne moguće implementacije na kraju zadaće. Čitavo rješenje (Visual Studio projekt) zajedno s eventualnim napomenama pohranite kao `prezime_ime.zip` (bez diakritičkih znakova) i predajte samo tu arhivu. Aplikacija treba biti implementirana koristeći sljedeće korisničke kontrole:

1. **datum**: sadrži tri `TextBox` (ili `NumericUpDown`) kontrole koje predstavljaju dan, mjesec i godinu. Defaultni konstruktor treba postaviti vrijednosti kontrola na vrijednosti iz svojstva `DateTime.Today`. Ako (i samo ako) korisnik upiše neke vrijednosti koje ne predstavljaju datum (npr. ako su upisane vrijednosti 30, 2, 2000), pozadina sve tri sadržane kontrole treba biti u nekoj nijansi crvene (u suprotnom, uobičajeno bijela). Provjera ne mora funkcionirati ispravno za datume prije 2000. niti nakon 2200. Štoviše, pretpostavljamo da se *nikad* ne unose datumi raniji od 1. 1. 2000. Interakcija s drugim kontrolama odvija se na sljedeći način:
 - Javno svojstvo (čitanje i pisanje) `DateTime` **vrijednost**. Ovo svojstvo vraća posljednje upisan ispravan datum. Takav datum sigurno postoji jer korisnička kontrola na početku sadrži podatke dobivene na temelju (ispravnog) `DateTime` objekta (trenutnog dana). Pridruživanje nove vrijednosti treba uzrokovati odgovarajuće popunjavanje tri sadržane kontrole.
 - Događaj **promjena** koji pozivatelju vraća informaciju o upisanom datumu, kad god korisnik kroz grafičko sučelje upiše novi ispravan datum. Događaj se može i ne mora pozivati kad se promjena dogodi na drugačiji način (npr. kroz kod). U svim ostalim situacijama, događaj se ne poziva.
2. **vremenski_raspon**: sadrži dvije `datum` korisničke kontrole, te pripadne `Label` kontrole s tekstovima “Početak:” i “Kraj:”. Interakcija s drugim kontrolama odvija se na sljedeći način:
 - Javno svojstvo (čitanje i pisanje) `Tuple<DateTime, DateTime>` **raspon**. Ovo svojstvo vraća par datuma dobivenih čitanjem **vrijednost** svojstava sadržanih kontrola. Pridruživanje novog para datuma pridružuje te datume svojstvima **vrijednost** sadržanih kontrola.
 - Događaj **promjena** koji pozivatelju vraća informaciju o upisanom paru datuma, kad god korisnik kroz grafičko sučelje upiše novi ispravan raspon. Događaj se može i ne mora pozivati kad se promjena dogodi na drugačiji način (npr. kroz kod). U svim ostalim situacijama, događaj se ne poziva. Raspon je ispravan ako su datumi smisleni, te je prvi datum manji ili jednak drugom datumu.

3. **graf**: ne sadrži druge kontrole. Interakcija s drugim kontrolama odvija se na sljedeći način:

- Javno svojstvo (čitanje i pisanje) `Tuple<DateTime, DateTime> raspon`. Ovo svojstvo vraća par datuma koji predstavljaju početnu i završnu točku prikazanu na grafu. Pridruživanje novog para datuma treba uzrokovati prikaz grafa funkcije sinus s jednom iscrtanom točkom na grafu po svakom danu sadržanom u rasponu (krajevi su uključeni). Ulazna vrijednost za funkciju sinus redni je broj datuma podijeljen s 10. Dane brojimo od 1. 1. 2000. (to je nulti dan). Npr. za 1. 1. 2001. ulazna vrijednost za sinus bit će 36.6 (2000. godina ima 366 dana, pa dani u 2000. godini imaju indekse od 0 do 365; stoga 1. 1. 2001. kao prvi sljedeći dan ima indeks 366). Treba nacrtati i *x*-os s crticama za svaki dan (ne treba ispisivati same datume ili njihove indekse). Nije potrebno crtati *y*-os.
- Događaj **dvoklik**, koji se poziva kad korisnik dvoklikne bilo gdje na području grafa.

Prilikom iscrtavanja (**Paint**), promjene veličine kontrole i pridruživanja nove vrijednosti svojstvu `raspon`, potrebno je ponovno iscrtati graf tako da pokrije većinu dostupnog područja kontrole (ostavite i nešto praznog prostora uz rubove zbog bolje preglednosti).

U gornji lijevi kut glavne forme dodajte kontrolu `vremenski_raspon`. Ispod nje treba doći kontrola **graf**, te se ona treba prilagođavati veličini prozora u svim smjerovima. Forma treba prije prikazivanja postaviti raspone obje kontrole na `raspon` koji počinje i završava trenutnim danom (`DateTime.Today`).

Potrebno je sinkronizirati ove dvije kontrole (`vremenski_raspon` i **graf**) na sljedeći način:

- Na događaj **promjena** kontrole `vremenski_raspon` potrebno je postaviti svojstvo `raspon` kontrole **graf** na dobivenu vrijednost.
- Na događaj **dvoklik** kontrole **graf** potrebno je postaviti svojstvo `raspon` kontrole **graf**, te svojstvo `vrijednost` kontrole `vremenski_raspon`, na vrijednost (1. 1. A., 31. 12. B.) gdje je A godina početka trenutnog raspona, a B godina završetka trenutnog raspona.

Napomene

- Korištenje javnih svojstava i događaja dio je zadaće, tj. nije samo prijedlog jedne moguće implementacije. Rješenje koje interakciju korisničkih kontrola ostvaruje na neki bitno drugačiji način (koje ne koristi javna svojstva i događaje) nosi najviše 5 bodova.
- Kod računanja koordinata za funkciju sinus uzmite u obzir da **Graphics** koristi koordinatni sustav u kojem su koordinate gornjeg lijevog kuta (0,0).
- Ako koristite `NumericUpDown` i uobičajeni događaj `ValueChanged`, možda ćete primijetiti da se on ne poziva na bilo kakvu promjenu teksta u kontroli, već tek kad kontrola izgubi fokus ili korisnik potvrdi unos. Tek se tada mijenja `Value` svojstvo. To je u redu, tj. nije nužno da je pozadinska boja kontrola odgovarajuća prije nego se pozove `ValueChanged`,
- Graf može biti raspršen (*scatter*), tj. dovoljno je nekako označiti svaku točku (to je napravljeno na snimci). Graf može biti i linijski, u tom slučaju susjedne točke povežite dužinom.

Primjer

[Snimka](#) jedne implementacije koja demonstrira sve funkcionalnosti osim dvoklika na graf.