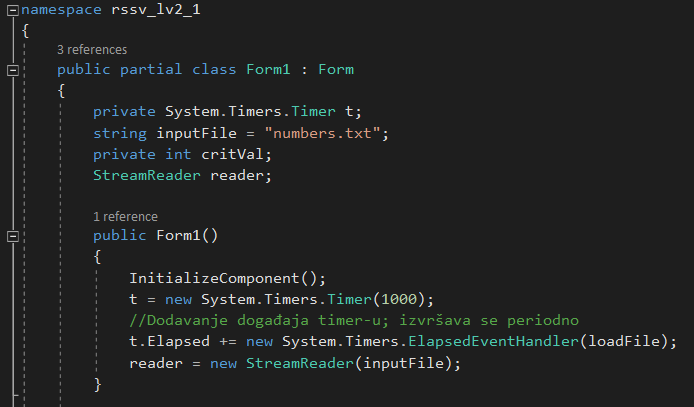
**KOLEGIJ:** Računalni sustavi stvarnog vremena

**PREDMET:** Izvještaj sa LV 2

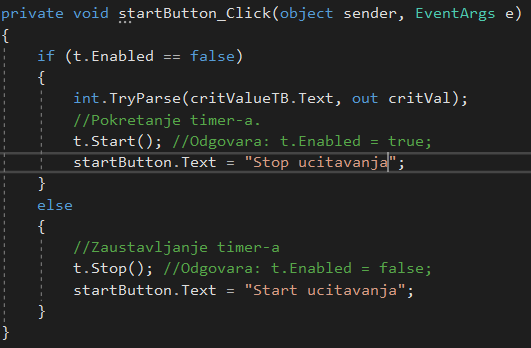
**Zadatak 1.**

Za prvi zadatak potrebno je napraviti aplikaciju koja će periodično čitati svaki broj iz zadane vrijednosti. Potrebno je napraviti funkcionalnost tako da korisnik na korisničkom sučelju ima buttone za pokretanje i zaustavljanje čitanja brojeva iz datoteke te input field u kojega će upisati željenu kritičnu vrijednost broja tako da će mu aplikacija u posebnom prozoru dati obavijest ukoliko je pročitani broj veći od kritične vrijednosti. Također, na korisničkom sučelju potrebno je napraviti button za izlazak iz aplikacije. Na sljedećoj slici prikazan je prvi dio koda koji se odnosi na deklaraciju timer-a **t**, StreamReadera **reader**,tekstualne datoteke iz koje ćemo čitati brojeve (ime datoteke je numbers.txt) te varijable u koju će se spremati iznos kritične vrijednosti. U StreamReaderu smo definirali da će se datoteka koju ćemo koristiti biti ona koju smo prethodno definirali u inputFile varijablu. Također definirali smo timer tako da će događaj izvršavati svaku sekundu (1000 u ms).



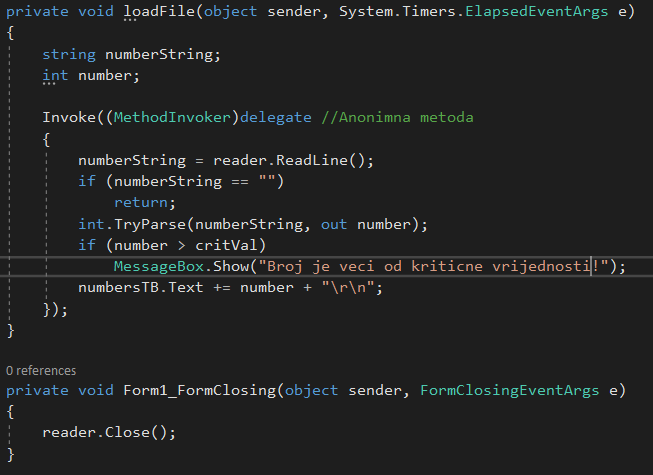
Slika 1.

Nastavak koda prikazan je na sljedećoj slici 2 iz koje vidimo kako smo definirali funkcionalnost klikom na button exitButton pomoću kojeg se izlazi iz aplikacije. Nadalje, sljedeći button je startButton odnosno onaj za pokretanje čitanja brojeva iz dane datoteke. Vidljivo je da tipka ima funkcionalnost da je tekst koji je na njoj u ovisnosti o vrijednost istinitosti za timer **t**, tako da će na tipki pisati „Start ucitavanja“ ukoliko nije pokrenut timer a ukoliko je pisat će „Stop ucitavanja“. Na ovaj način pomoću jedne tipke ćemo regulirati pokretanje i zaustavljanje očitavanja brojeva iz datoteke.



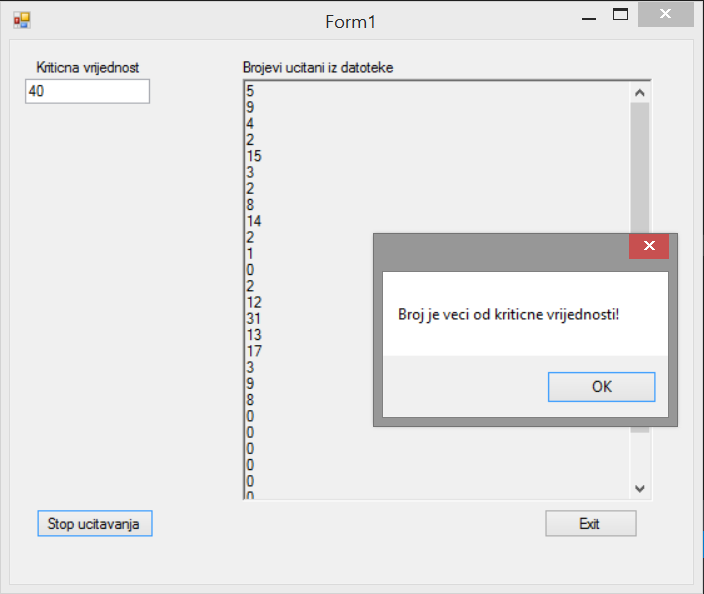
Slika 2.

Na sljedećoj slici prikazane su petlje i metode kojima čitamo i provjeravamo vrijednost učitanog broja te u slučaju da je broj veći od zadane kritične vrijednosti ispisuje je MessageBox sa odgovarajućim tekstom. Koristi se metoda Invoke() koja izvršava delegata(pokazivac) na nit koja je vlasnik kontrola prozora. Na slici se također vidi se nakon zatvaranje forma gasi StreamReader reader.



Slika 3.

Na narednoj slici prikazan je konačni izgled i funkcionalnost aplikacije za 1. Zadatak.

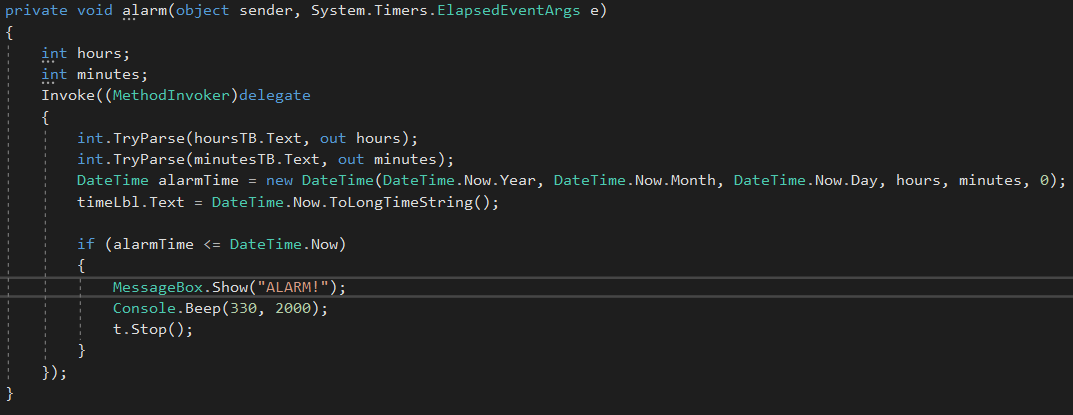


Slika 4.

**Zadatak 2.**

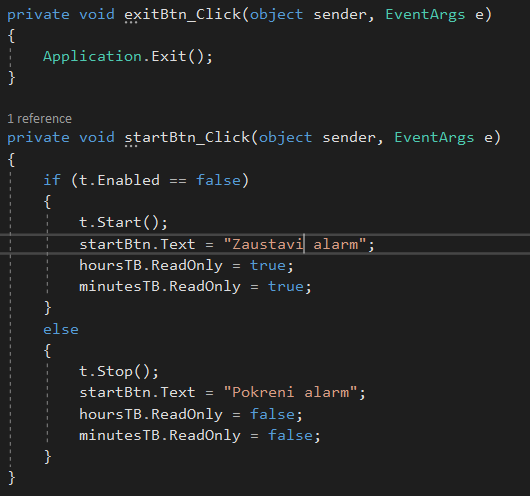
Drugi zadatak bio napraviti program koji bi simulirao rad alarma. Kroz korisničko sučelje potrebno je omogućiti zadavanje vrijeme oglašavanja alarma.

Na slici 5. prikazano je deklariranje timera na isti način kao u prošlom zadatku. Nakon toga slijedi kreiranje funkcionalnosti za događaj u kojemu deklariramo vrijedosti za odabrane sate i minute u kojima želimo aktivirati alarm. Vidljivo je da ukoliko kada vrijeme alarma dođe na trenutnu vremensku vrijednost, timer se zaustavlja te daju zvučni signal pomoću Beep() klase. Godina, mjesec i dan uzima se od današnjeg dana dok mi sami definiramo sate i minute paljenja alarma.



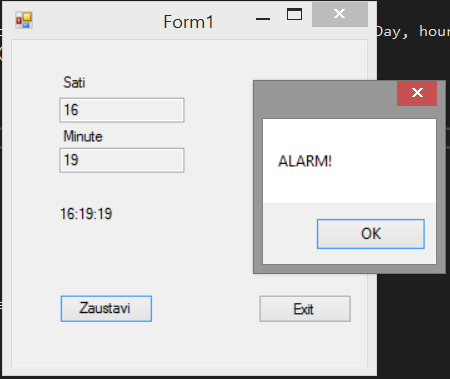
Slika 5.

Na slici 6. Prikazana je funkcionalnost tipki za pokretanje i zaustavljanje alarma te za gašenje programa. Ovdje se kao i u prošlom zadatku provjerava stanje timera te se na temelju toga određuje natpis na button-u. Izlazak iz aplikacije napravljen je kao i u prošlom zadatku.



Slika 6.

Alarm se oglašava Beep() tonom dok se za notifikaciju prikaže prozor na kojemu također piše Alarm. Konačni program prikazan je na slici 7.



Slika 7.

**Vanja Matoković,**

**Osijek, 05. lipnja 2017. godine**