

# Projektna naloga IZPIT RPA

Projekt poimenujte **PriimeklmeRPA** in ga shranite v mapo z **enakim** imenom.

Ustvarite projekt za vodenje evidence pekarne. Podatki naj bodo shranjeni v bazi **Pekarna**, ki naj se nahaja v mapi <http://uranic.tsckr.si/Lokalne%20baze/> (arhiv **Pekarna.zip**). Baza ima dve tabeli: v tabeli **Prodaja** je seznam prodanih pekarskih izdelkov, v tabeli **Izdelki** pa so izdelki, ki se prodajajo v pekarni, zraven pa še njihova cena. Strukturi obeh datotek:

**Izdelki: Izdelek:** nvarchar(50), ključno polje, Allow Nulls (Ne)

**Cena:** float, Allow Nulls (Ne)

**Prodaja: Zaporedna:** int, ključno polje, Autoincrement, AllowNulls (Ne)

**Izdelek,** nvarchar(50), Allow Nulls (Ne)

**Kolicina:** int, Allow Nulls (Ne)

**Cena:** float, Allow Nulls (Ne)

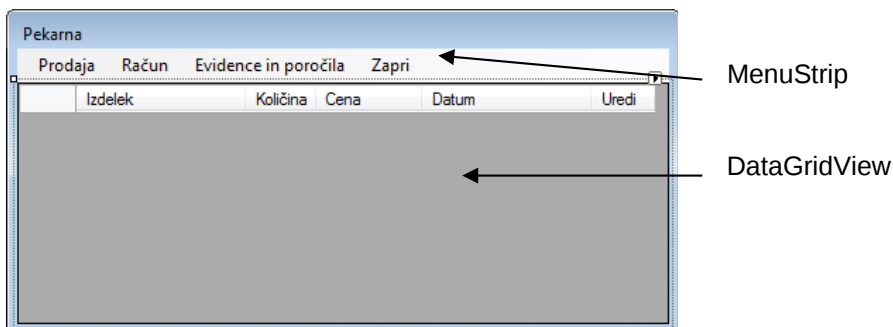
**Datum:** DateTime, Allow Nulls (Ne)

Obe tabeli že imata nekaj zapisov.

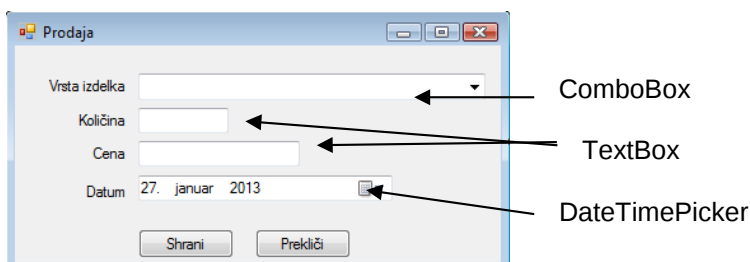
Na glavnem obrazcu projekta naj bo meni z naslednjimi postavkami:

**Prodaja, Račun, Evidence in Poročila in Zapri**

Na glavnem obrazcu naj bo še gradnik **DataGridView**, v katerem ustvarite stolpce tako, kot prikazuje slika: vsi stolpci so tipa **DataGridViewTextBoxColumn**, razen stolpca **Uredi**, ki je tipa **DataGridViewButtonColumn**. Numerični stolpci naj imajo desno poravnavo. Vsi stolpci naj bodo **ReadOnly**, neposredno dodajanje nove vrstice in urejanje poljubne vrstice pa naj ne bo dovoljeno. (5 točk)



Ob kliku na možnost **Račun** v glavnem meniju naj se odpre obrazec **Prodaja**:



Gradnik **ComboBox** na tem obrazcu poveži s tabelo **Izdelki** (**DisplayMember** je vrsta izdelka, **ValueMember** je cena izdelka). Prodajalec s pomočjo spustnega seznama izbere vrsto izdelka. Po izbiri vrste izdelka naj se v polje **Cena** zapiše cena tega izdelka, v gradnik **Količina** pa privzeta količina 1. (5 točk)

Prodajalec naj ima na obrazcu **Prodaja** možnost spremembe količine in datuma (*ne pa tudi cene*). Ob kliku na gumb **Shrani** naj se podatki z obrazca **Prodaja** zapišejo v novo vrstico gradnika **DataGridView**. Ob kliku na gumb **Prekliči** naj se obrazec **Prodaja** le zapre. Pred zapisom podatkov v gradnik **DataGridView** preveri pravilnost vnosa podatkov (uporabiš lahko gradnik **ErrorProvider**,

odzivne metode dogodka **KeyPress**, ali pa enostaven odločitveni stavek, s katerim preveriš pravilnost vnosov). **(15 točk)**

S klikom na gumb **Uredi** gradnika **DataGridView** naj se podatki izbrane vrstice prenesejo nazaj na obrazec **Prodaja**, kjer jih prodajalec lahko uredi in ponovno shrani nazaj na isto mesto. **(5 točk)**

Klik na gumb **Račun** v glavnem meniju pomeni, da naj se celotna vsebina gradnika **DataGridView** doda v tabelo **Prodaja** baze **Pekarna**, nato pa naj se vrstice gradnika **DataGridView** pobrišejo. Obenem naj se skupni seznam prenesenih artiklov, ter končni skupni znesek izpiše na tiskalniku – oblika izpisa je lahko poljubna, brez posebnega oblikovanja npr.:

**(20 točk)**

**Artikli: Žemlja 2x, Rogljič 1x, Kruh Beli 0,5x ..... Skupni znesek = 216 EUR**

Če dela s tiskalnikom ne poznate, lahko končni izpis prikažete v sporočilnem oknu (**odbitek 5 točk**).

Možnost **Evidence in Poročila** v meniju glavnega obrazca naj vsebuje

- **Skupna prodaja:** ob kliku naj se odpre nov obrazec, na katerem naj bo prikazana vsebina tabele **Prodaja** baze **Pekarna**; **(3 točke)**
- **Skupni znesek izdelkov:** v sporočilnem oknu naj se izpiše skupni znesek izdelkov, ki so trenutno v gradniku **DataGridView**; **(3 točke)**
- **Skupna količina vseh prodanih artiklov:** izračunajte in izpišite skupno količino vseh izdelkov iz datoteke **Prodaja.txt**. Rezultat izpišite v sporočilnem oknu. **(4 točke)**

**SKUPAJ: 60 točk**

**Izpit: 30 – 5, 31 - 36 = 6, 37 – 42 = 7, 43 – 48=8, 49 – 54 = 9, 55 – 60 =10**

**Seminarska: 0 – 20 = 5, 21 – 24 = 6, 25 – 28 = 7, 29 – 32 = 8, 33 – 36 = 9, 37 – 40 = 10**

**Skupaj: 50 - 60 = 6, 61 – 70 = 7, 71 – 80 = 8, 81 – 90 = 9, 91 – 100 = 10**