

POČÍTAČOVÉ CVIČENÍ 8

Cílem cvičení je procvičit si práci se strukturami a dynamickými proměnnými s navázáním na formulářovou aplikaci pro Windows.

1. Příklad vychází z příkladu **C07_1.cpp** pro správu databáze zelinářství. Cílem je upravit funkce pro vytváření dynamických struktur skladu ovoce a zeleniny a zakomponovat je do předpřipraveného nadefinovaného formuláře s potřebnými ovládacími prvky, viz. projekt **C08_1.sln**. Řešení prodeje není v této úloze implantováno. Upravte, případně doplňte: funkci pro správu databáze skladu, tj. funkce pro inicializaci záznamů (na začátku musí být **status** nastaven na free, tj. symbolická znaková konstanta **'f'**), např. se jménem **Rec_Init()**, funkci pro naplnění skladu vybranými položkami podobně jako v předchozím příkladě, např. **Init_Store()**, dále pak funkce pro přidání nového zboží **Add_New_Item()** a funkce pro doplnění (přidání) množství již zavedeného zboží **Add_Goods()**. Užitečná bude i funkce pro vyhledání volného, resp. obsazeného, záznamu (kódu) ve skladu **Get_Free_Code()** a funkce pro výběr všech údajů z daného záznamu s generováním řetězce pro zobrazení v ovládacím prvku typu **ListBox** např. s názvem **Get_Item_From_Store()**. Z těchto funkcí sestavte knihovnu např. s názvem **greengrocery.h**. Funkce z této knihovny pak budou volány při zpracování událostí příslušných ovládacích prvků z formuláře. V této fázi je vhodné projít si vlastní formulář aplikace a jednotlivé ovládací prvky. V aplikaci jsou již připraveny funkce **Show_Store_Form()**, **Add_New_Item_Form()** a **Add_Goods_Form()**, které definují vzhled formuláře pro jednotlivé úkoly a které jsou již součástí vybraných funkcí pro ošetření událostí a lze je bezprostředně vyzkoušet. Pro orientaci v programu je do zdrojového kódu formuláře přidána globální znaková proměnná **app_mode**, která definuje aktuální stav aplikace (**'A'** znamená základní stav, kdy se čeká na volbu požadované operace, **'I'** je stav, kdy se vkládá nová položka do skladu a **'G'** je stav, kdy se přidává dané množství již zavedené položky). Nyní je třeba doplnit zdrojový kód formuláře o potřebný kód k obsluze ovládacích prvků. Výhodné je zapouzdření vybraných částí kódu do samostatných funkcí: např. pro výpis obsahu skladu v **ListBoxu LB_Main** nebo obsazených či neobsazených kódů skladu do **ComboBoxu CB_Code**. Pro výpis kompletního jednoho záznamu do **ListBoxu LB_Main** je třeba vytvořit řetězec (kde bude kód zboží, druh zboží, kolik je ho na skladě a kolik stojí) nejlépe pomocí funkce tisku do řetězce **sprintf()** z knihovny **stdio.h** a vytvořeny řetězec vložit do proměnné **item** (viz. dále). Příklad výpisu je patrný z obrázku 1.

Poznámky:

Přidání položky v **ListBoxu** nebo **ComboBoxu** se jménem **object**

- a) textový řetězec **str**

```
object->Items->Add(String::Format(gcnew System::String(str)));
```

- b) číslo typu **int num**

```
object->Items->Add(String::Format("{0}", num));
```

Mazání **n-té** položky v **ListBoxu** nebo **ComboBoxu** se jménem **object**

```
object->Items->RemoveAt(n);
```

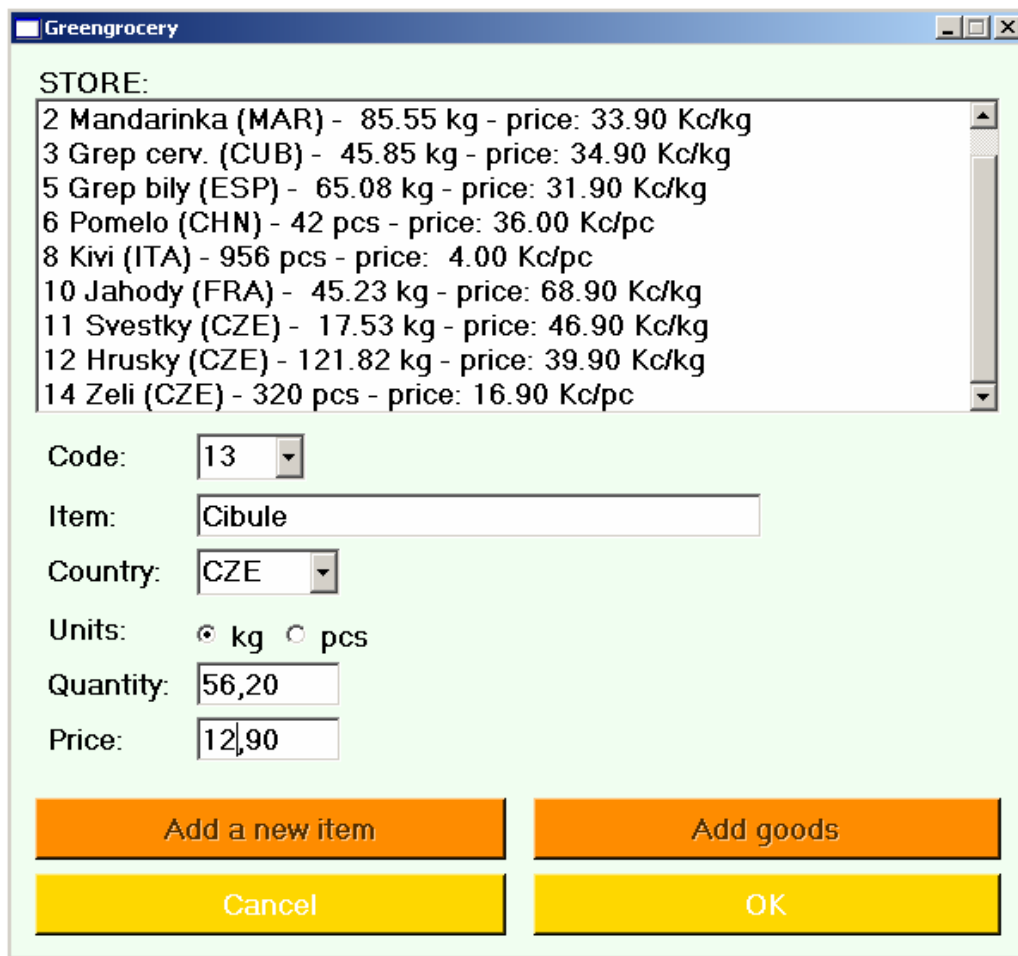
Doporučený postup řešení:

- a) Upravit a sestavit funkce pro knihovnu **greengrocery.h** a přidat ji do zdrojového kódu formuláře.

- b) Sestavit kód (nejlépe jako funkci) v zdrojovém kódu formuláře pro výpis obsahu skladu a otestovat funkčnost.
- c) Sestavit kód (nejlépe jako funkci) ve zdrojovém kódu formuláře pro výpis volných kódů ve skladu a obsazených kódů ve skladu a otestovat funkčnost výpisem v **ComboBoxu CB_Code**. Funkce budou vhodné pro operace přidání množství již existujícího zboží ve skladu (výběr z obsazených kódů v **CB_Code**) a pro přidání nového zboží (výběr z volných kódů).
- d) Sestavit kód ve zdrojovém kódu formuláře pro navýšení množství již existujícího zboží ve skladu a otestovat. Při novém zobrazení obsahu skladu musí být tato položka o dané množství navýšena.
- e) Sestavit kód ve zdrojovém kódu formuláře pro přidání nového zboží se všemi údaji s vytvořením nového záznamu (dynamicky alokovaného v paměti) a otestovat. Při novém zobrazení obsahu skladu musí být toto zboží se všemi položkami zobrazeno.

2. Neveřejný příklad zadaný cvičícím.

HODNOCENÍ: Za splnění prvního příkladu 1 bod, za splnění druhého příkladu 1 bod.



The screenshot shows a window titled "Greengrocery". Inside, there is a list of items under the heading "STORE:". The items are:

- 2 Mandarinka (MAR) - 85.55 kg - price: 33.90 Kc/kg
- 3 Grep cerv. (CUB) - 45.85 kg - price: 34.90 Kc/kg
- 5 Grep bily (ESP) - 65.08 kg - price: 31.90 Kc/kg
- 6 Pomelo (CHN) - 42 pcs - price: 36.00 Kc/pc
- 8 Kivi (ITA) - 956 pcs - price: 4.00 Kc/pc
- 10 Jahody (FRA) - 45.23 kg - price: 68.90 Kc/kg
- 11 Svestky (CZE) - 17.53 kg - price: 46.90 Kc/kg
- 12 Hrusky (CZE) - 121.82 kg - price: 39.90 Kc/kg
- 14 Zeli (CZE) - 320 pcs - price: 16.90 Kc/pc

Below the list, there are input fields for adding a new item:

- Code: 13 (dropdown menu)
- Item: Cibule (text input)
- Country: CZE (dropdown menu)
- Units: ☒ kg ☐ pcs
- Quantity: 56,20 (text input)
- Price: 12,90 (text input)

At the bottom, there are four buttons: "Add a new item" (orange), "Add goods" (orange), "Cancel" (yellow), and "OK" (yellow).

Obr. 1. Příklad zobrazení výsledků příkladu ve formuláři.