Dílenská praxe

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **A4** | 4. Snímač čárkového kódu | | | |
| Macháček Daniel | |  | 1/7 | Známka: |
| 30.11.2016 | | Datum odevzdání: | 4.1.2017 | Odevzdáno: |

Zadání:

Zpracujte program v programovacím jazyce C# simulující provoz jednoduchého pokladního místa tak, aby obsahoval nejméně tyto funkce:

1. Ke vstupu dat použijte USB snímač čárkového kódu (zboží i nakupujících).
2. Databázi sledovaných položek (zboží i nakupujících) simulujte externími textovými soubory ve **formátu CSV.**
3. Vytiskněte jednotlivé seznamy prodaného zboží. Tisk simulujte zápisem do textového souboru.
4. Umožněte vytisknout souhrnný přehled pohybu zboží pokladním místem a to jak po položkách zboží, tak i po zákaznících. Tisk simulujte zápisem do textového souboru.

**Postup ovládání:**

1. Jako první, je třeba načíst kód zákazníka (Platné kódy naleznete níže).
2. Klikněte do prvního textBoxu, nahrajte kód a potvrďte tlačítkem „Načti zákazníka“, jméno příslušného zákazníka se zobrazí namísto textu „labeljmeno“.
3. Nyní můžete nakupovat zboží, překlikněte tedy do druhého textBoxu, načtěte kód produktu a stiskněte „Načti produkt“.
4. Název a cena produktu se opět zobrazí vedle a tlačítko „Načti zákazníka“ přestane být aktivní.
5. Tlačítko se opět stane aktivní, až aktuální nakupující dokončí nákup. Takže pokud už jste přidali všechny produkty, stiskněte Zaplatit (Výsledná cena se zobrazuje na účtence z celého nákupu, ale je ji nutno manuálně otevřít)

* Také se vyčistili oba textBoxy a můžete naskenovat nákup dalšího zákazníka.

1. Cteckakodu\Cteckakodu\bin\Debug\uctenka.txt
2. Tlačítko „Přehled“ funguje kdykoli nezávisle na pozici programu, ovšem opět je nutno si přehled manuálně otevřít (Cesta je stejná, jen soubor nese jiný název – souhrn.txt)

Závěr:

Tentokráte jsem si již dal pozor na chybu způsobenou vynecháním neznámého souboru, díky němuž nešel program spustit odjinud než zdrojové složky mého počítače. Program splňuje všechny body zadání a je plně funkční avšak samotná čtečka vyzkoušena nebyla. Nemyslím si, že by to mělo znamenat problém. Na druhou stranu program postrádá spoustu ošetření například vůči zadání špatného kódu. Řešení pro výpis „Přehledu“ také nebylo zcela nejlepší. V případě velkého skladu s mnoha produkty by byl program zbytečně dlouhý. Řešením by pravděpodobně bylo požitím „foreach“, ale nezbyl mi čas, abych tuto možnost sám vyzkoušel.

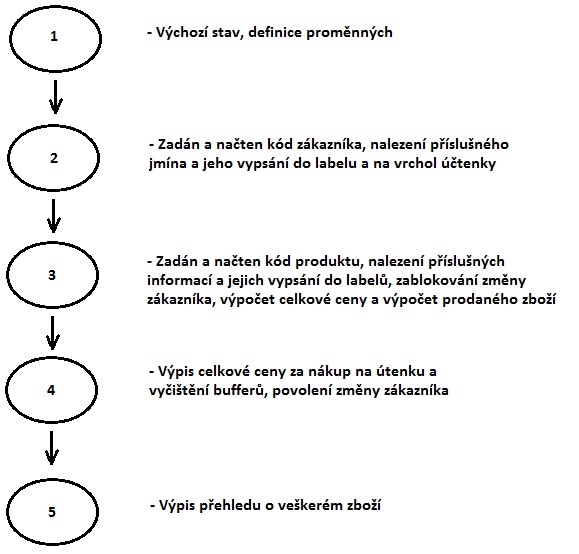
Platné kódy:

Přílohy:

* Výpis programu – 4 strany
* Stavový diagram – 1 strana

**Stavový Diagram:**



**Výpis programu:**

using System;

using System.Collections.Generic;

using System.ComponentModel;

using System.Data;

using System.Drawing;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows.Forms;

using System.IO;

namespace Cteckakodu

{

public partial class Form1 : Form

{

public Form1()

{

InitializeComponent();

}

string Jmeno; //Definice proměnných

string Nazev;

string Cena;

string Kod;

string Zkod;

string Pkod;

string Read;

float Vypocet;

float CenaZaNakup;

int P = 0;

int M = 0;

int A = 0;

int F = 0;

int Pe = 0;

int W = 0;

private void NactiZ\_Click(object sender, EventArgs e) //Tlačítko - Načtení zákazníka

{

Zkod = textBoxzakaznik.Text; //Přečtu text z textboxu a uložím jej do Zkod

using (StreamReader ZR = new StreamReader(@"Zakaznici.csv"))

//Povolím Načítání Zakaznici.csv

{

while (Zkod != Kod) //Vyhledávám zadaný kód v csv souboru

{

Read = ZR.ReadLine(); //Přečtu řádek a uložím jej do Read

string[] separate = Read.Split(';'); //Rozdělím ho podle středníků

Kod = separate[0]; //První část uložím na nultou hodnotu pole

Jmeno = separate[1]; //Druhou na první hodnotu pole

}

}

using (StreamWriter ZW = new StreamWriter(@"uctenka.txt"))

//Povolím zápis do vytvořeného adresáře .txt

{

ZW.WriteLine(" " + Jmeno); //Navrh účtenky vypíšu jméno zákazníka

ZW.Flush(); //Vyčistí buffer

}

labeljmeno.Text = Jmeno; //V labelu pro jméno zobrazí jméno aktuálního zákazníka

}

private void NactiP\_Click(object sender, EventArgs e) //Tlačítko - Načtení produktu

{

NactiZ.Enabled = false; //Deaktivuje tlačítko pro načtení zákazníka

Pkod = textBoxprodukt.Text;

using (StreamReader PR = new StreamReader(@"Produkty.csv"))

{

while (Pkod != Kod)

{

Read = PR.ReadLine();

string[] separate = Read.Split(';');

Kod = separate[0];

Nazev = separate[1];

Cena = separate[2];

}

}

using (StreamWriter PW = new StreamWriter(@"uctenka.txt", true))   
 //Povolí zápis do vytvořeného adresáře .txt s možností přepisovat

{

PW.WriteLine(" " + Nazev + " " + Cena);

PW.Flush();

}

labelprodukt.Text = Nazev;

labelcena.Text = Cena;

if (Single.TryParse(Cena, out Vypocet)) { /\*Ono to funguje\*/ } //Konvertuje cenu produktu

else { MessageBox.Show("Nezdařil se převod ceny.", "Chyba", MessageBoxButtons.OK,   
 MessageBoxIcon.Error);

} //Chybová hláška kdyby hodnota nešla převést

CenaZaNakup = CenaZaNakup + Vypocet; //Výpočet celkové ceny nákupu

if (Kod == "4681234687246") { P = P + 1; } //Výpočty pro přehled prodaného zboží

if (Kod == "135897156716") { M = M + 1; }

if (Kod == "646497521399") { A = A + 1; }

if (Kod == "987316520068") { F = F + 1; }

if (Kod == "404968051323") { Pe = Pe + 1; }

if (Kod == "330685046977") { W = W + 1; }

}

private void Zaplatit\_Click(object sender, EventArgs e) //Tlačítko - Účtenka

{

using (StreamWriter CW = new StreamWriter(@"uctenka.txt", true))

{

CW.WriteLine("Celkem: " + CenaZaNakup + " Kč");

CW.Flush();

}

textBoxzakaznik.Clear();

textBoxprodukt.Clear();

NactiZ.Enabled = true; //Aktivuje tlačítko pro načtení zákazníka

}

private void Prehled\_Click(object sender, EventArgs e) //Tlačítko - Přehled prodaného zboží

{

using (StreamWriter SW = new StreamWriter(@"souhrn.txt", true))

{

SW.WriteLine("Perla Plus: " + P + "krát");

SW.WriteLine("Mattoni 1,5l PET: " + M + "krát");

SW.WriteLine("Ananas - velký: " + A + "krát");

SW.WriteLine("Fern.Stock Citrus: " + F + "krát");

SW.WriteLine("Vícezrnné pečivo: " + Pe + "krát");

SW.WriteLine("Houska Wiking: " + W + "krát");

SW.Flush();

}

}

}

}