

```
1 using System;
2 using System.Collections.Generic;
3 using System.ComponentModel;
4 using System.Data;
5 using System.Drawing;
6 using System.Linq;
7 using System.Text;
8 using System.Threading.Tasks;
9 using System.Windows.Forms;
10 using System.IO;
11
12 namespace knapsackplanner
13 {
14     public partial class Form1 : Form
15     {
16
17         private BindingList<Item> _items;
18         private BindingList<Container> _containers;
19         private enum ItemSelected { Item, Container, ContainedItem};
20         private ItemSelected _currentSelection;
21         public Form1()
22         {
23             InitializeComponent();
24             _items = new BindingList<Item>();
25             _containers = new BindingList<Container>();
26             _dataGridViewItems.DataSource = _items;
27             _dataGridViewContainers.DataSource = _containers;
28
29             StreamReader reader;
30             string[] lineRead = new string[5];
31
32
33             try
34             {
35                 //open file
36                 reader = File.OpenText("knapsack.csv");
37
38
39
40                 //run through all lines
41                 while (!reader.EndOfStream)
42                 {
43                     //read next line
44                     lineRead = reader.ReadLine().Split(',');
45                     if (lineRead[0] == "ITEM")
46                     {
47                         _items.Add(new Item(lineRead[1], double.Parse(lineRead
48 [2]), double.Parse(lineRead[3]), decimal.Parse(lineRead
49 [4])));
50                     }
51                     else if (lineRead[0] == "CONTAINER")
52                     {
53                         _containers.Add(new Container(lineRead[1],
54 double.Parse(lineRead[2]), double.Parse(lineRead[3])));
55                     }
56                 }
57             }
58         }
59     }
60 }
```

```
54         reader.Close();
55     }
56     catch (Exception ex)
57     {
58         MessageBox.Show(ex.Message);
59     }
60     _dataGridViewStoredItems.DataSource = _containers[0];
61 }
62
63 internal Container Container
64 {
65     get => default;
66     set
67     {
68     }
69 }
70
71 private void _buttonMoveTo_Click(object sender, EventArgs e)
72 {
73     DataGridViewCell containerCell =
74         _dataGridViewContainers.CurrentCell;
75     if (containerCell == null)
76     {
77         MessageBox.Show("No Container available!");
78         return;
79     }
80     DataGridViewCell itemCell = _dataGridViewItems.CurrentCell;
81     if (itemCell == null)
82     {
83         MessageBox.Show("No Item available!");
84         return;
85     }
86     int containerIndex = containerCell.RowIndex;
87     int itemIndex = itemCell.RowIndex;
88     Item item = _items[itemIndex];
89     Container container = _containers[containerIndex];
90     double totalWeight = 0;
91     double totalVolume = 0;
92     foreach (Item item1 in container.ItemsContained)
93     {
94         totalWeight += item1.Weight;
95         totalVolume += item1.Volume;
96     }
97     if ((totalWeight+item.Weight)>container.MaxWeight || (totalVolume
98         + item.Volume) > container.MaxVolume)
99     {
100         MessageBox.Show("Exceeds max weight/volume!");
101         return;
102     }
103     container.ItemsContained.Add(item);
104     _items.Remove(item);
105     container.Update();
106     _dataGridViewContainers.DataSource = null;
107     _dataGridViewContainers.DataSource = _containers;
```

```
108
109     private void _buttonMoveFrom_Click(object sender, EventArgs e)
110     {
111         DataGridViewCell containerCell =
112             _dataGridViewContainers.CurrentCell;
113         if (containerCell == null)
114         {
115             MessageBox.Show("No Container available!");
116             return;
117         }
118         DataGridViewCell itemCell = _dataGridViewStoredItems.CurrentCell;
119         if (itemCell == null)
120         {
121             MessageBox.Show("No Item available!");
122             return;
123         }
124         int containerIndex = containerCell.RowIndex;
125         int itemIndex = itemCell.RowIndex;
126         Container container = _containers[containerIndex];
127         Item item = container.ItemsContained[itemIndex];
128         container.ItemsContained.Remove(item);
129         _items.Add(item);
130         container.Update();
131         _dataGridViewContainers.DataSource = null;
132         _dataGridViewContainers.DataSource = _containers;
133     }
134
135
136     private void _dataGridViewContainers_SelectionChanged(object sender,
137         EventArgs e)
138     {
139         DataGridViewCell cell = _dataGridViewContainers.CurrentCell;
140         if (cell != null)
141         {
142             _currentSelection = ItemSelected.Container;
143             int containerIndex = cell.RowIndex;
144             Container container = _containers[containerIndex];
145             container.Bind(_dataGridViewStoredItems);
146
147             _textBoxName.Text = container.Name;
148             _numericUpDownWeight.Value = (decimal)container.MaxWeight;
149             _numericUpDownVolume.Value = (decimal)container.MaxVolume;
150             _numericUpDownValue.Value = 0;
151         }
152     }
153
154     private void _dataGridViewItems_SelectionChanged(object sender,
155         EventArgs e)
156     {
157         DataGridViewCell cell = _dataGridViewItems.CurrentCell;
158         if (cell != null)
159         {
160             _currentSelection = ItemSelected.Item;
161             int itemIndex = cell.RowIndex;
```

```
161         Item item = _items[itemIndex];
162
163         _textBoxName.Text = item.Name;
164         _numericUpDownWeight.Value = (decimal)item.Weight;
165         _numericUpDownVolume.Value = (decimal)item.Volume;
166         _numericUpDownValue.Value = item.Value;
167
168     }
169 }
170
171 private void _dataGridViewStoredItems_SelectionChanged(object sender, EventArgs e)
172 {
173     DataGridViewCell cell = _dataGridViewStoredItems.CurrentCell;
174     if (cell != null)
175     {
176         _currentSelection = ItemSelected.ContainedItem;
177         DataGridViewCell containerCell =
178             _dataGridViewContainers.CurrentCell;
179         DataGridViewCell itemCell =
180             _dataGridViewStoredItems.CurrentCell;
181         int containerIndex = containerCell.RowIndex;
182         Container container = _containers[containerIndex];
183         int itemIndex = itemCell.RowIndex;
184         if (container.ItemsContained.Count > 0)
185         {
186             Item item = container.ItemsContained[itemIndex];
187             _textBoxName.Text = item.Name;
188             _numericUpDownWeight.Value = (decimal)item.Weight;
189             _numericUpDownVolume.Value = (decimal)item.Volume;
190             _numericUpDownValue.Value = item.Value;
191         }
192     }
193 }
194
195 private void buttonEmpty_Click(object sender, EventArgs e)
196 {
197     DataGridViewCell containerCell =
198         _dataGridViewContainers.CurrentCell;
199     if (containerCell == null)
200     {
201         MessageBox.Show("No Container available!");
202         return;
203     }
204     int containerIndex = containerCell.RowIndex;
205     Container container = _containers[containerIndex];
206     for(int i = container.ItemsContained.Count - 1; i>=0; i--)
207     {
208
209         _items.Add(container.ItemsContained[i]);
210         container.ItemsContained.Remove(container.ItemsContained[i]);
211
212     }
```

```
213     }
214     container.Update();
215     _dataGridViewContainers.DataSource = null;
216     _dataGridViewContainers.DataSource = _containers;
217 }
218
219 private void _buttonAddItem_Click(object sender, EventArgs e)
220 {
221     _items.Add(new Item(_textBoxName.Text, (double)
222         _numericUpDownWeight.Value, (double)_numericUpDownVolume.Value,
223         _numericUpDownValue.Value));
224 }
225
226 private void _buttonDelete_Click(object sender, EventArgs e)
227 {
228     DataGridViewCell itemCell = _dataGridViewItems.CurrentCell;
229     DataGridViewCell storedItemCell =
230         _dataGridViewStoredItems.CurrentCell;
231     DataGridViewCell containerCell =
232         _dataGridViewContainers.CurrentCell;
233
234     switch (_currentSelection)
235     {
236     case ItemSelected.Item:
237         if (itemCell != null)
238         {
239             int itemIndex = itemCell.RowIndex;
240             Item item = _items[itemIndex];
241             _items.Remove(item);
242         }
243         break;
244
245     case ItemSelected.ContainedItem:
246         if (storedItemCell != null)
247         {
248             int containerIndex = containerCell.RowIndex;
249             Container container = _containers[containerIndex];
250             int itemIndex = storedItemCell.RowIndex;
251             if (container.ItemsContained.Count > 0)
252             {
253                 Item item = container.ItemsContained[itemIndex];
254                 container.ItemsContained.Remove(item);
255                 container.Update();
256                 _dataGridViewContainers.DataSource = null;
257                 _dataGridViewContainers.DataSource = _containers;
258             }
259         }
260         break;
261
262     case ItemSelected.Container:
263         if (containerCell != null)
264         {
```

```
265         int containerIndex = containerCell.RowIndex;
266         Container container = _containers[containerIndex];
267         _containers.Remove(container);
268     }
269     break;
270
271     default:
272         break;
273 }
274 }
275
276 private void _buttonAddContainer_Click(object sender, EventArgs e)
277 {
278     _containers.Add(new Container(_textBoxName.Text, (double)
279         _numericUpDownWeight.Value, (double)
280         _numericUpDownVolume.Value));
281
282 private void _buttonEdit_Click(object sender, EventArgs e)
283 {
284     DataGridViewCell itemCell = _dataGridViewItems.CurrentCell;
285     DataGridViewCell storedItemCell =
286         _dataGridViewStoredItems.CurrentCell;
287     DataGridViewCell containerCell =
288         _dataGridViewContainers.CurrentCell;
289
290     switch (_currentSelection)
291     {
292     case ItemSelected.Item:
293         if (itemCell != null)
294         {
295             int itemIndex = itemCell.RowIndex;
296             Item item = new Item(_textBoxName.Text, (double)
297                 _numericUpDownWeight.Value, (double)
298                 _numericUpDownVolume.Value, _numericUpDownValue.Value);
299             _items[itemIndex] = item;
300         }
301         break;
302
303     case ItemSelected.ContainedItem:
304         MessageBox.Show("Items cannot be edited while in a
305             container");
306         break;
307
308     case ItemSelected.Container:
309         if (containerCell != null)
310         {
311             int containerIndex = containerCell.RowIndex;
312             Container container = new Container(_textBoxName.Text,
313                 (double)_numericUpDownWeight.Value, (double)
314                 _numericUpDownVolume.Value);
315             _containers[containerIndex] = container;
316         }
317         break;
```

```
312
313         default:
314             break;
315     }
316 }
317 }
318 }
319
```