グレープシティ株式会社

Excel 帳票作成スタートガイド

DioDocs for Excel で Excel 帳票を作成

2022 年 1 月 11 日

目次

はじめに 3
Excel 帳票を作成する 2 つの方法 3
データの埋め込み（コード） 3
データの埋め込み（テンプレート構文） 4
テンプレート構文の概要 6
テンプレート構文の詳細 7
データソース 7
テンプレートフィールド 8
データフィールド 8
関数フィールド 8
式フィールド 8
数式フィールド 9
シート名フィールド 9
テンプレートプロパティ 10
セルコンテキスト 10
セル拡張 11
グループ化 12
セル範囲 14
ソート 15
ページ区切り 17
塗りつぶしモード 19
請求書の作成 22
開発環境の準備 22
出力結果 23
請求書（Excel ファイル出力） 23
請求書（PDF ファイル出力） 25
作成手順 26
1. コンソールアプリケーションの作成 26
2. DioDocs のパッケージを追加 26
3. データソースの準備 27
4. データソースの追加 31
5. 帳票テンプレートの作成 32
6. テンプレート処理の実行、Excel／PDF ファイル出力 35
アプリケーションの実行 36
請求書のカスタマイズ 37
ケース 1：可変明細の出力 37
帳票テンプレートの変更 37
出力結果（Excel ファイル） 38
ケース 2：請求先を単一シートにまとめる 39

帳票テンプレートの変更 39
出力結果（Excel ファイル） 40
出力結果（PDF ファイル） 41
ケース 3：備考欄とバーコードの追加 42
帳票テンプレートの変更 43
出力結果（Excel ファイル） 44
出力結果（PDF ファイル） 45

はじめに

本資料では主に初心者の方を対象に DioDocs for Excel（以下 DioDocs）を使った帳票作成方法を解説します。前半は基本的な使い方やテンプレート構文について、後半は実際に請求書を作成する手順を紹介します。

Excel 帳票を作成する 2 つの方法

DioDocs を使用した Excel 帳票作成の基本は、Microsoft Excel で作成した帳票テンプレートを DioDocs で読み込み、対象となるワークブックの個々のセルにデータを埋め込んでいくというものです。データを埋め込む処理はコード（C#、VB.NET）でひとつずつ記載する方法と、テンプレート構文という機能を使用して自動化する方法があります。

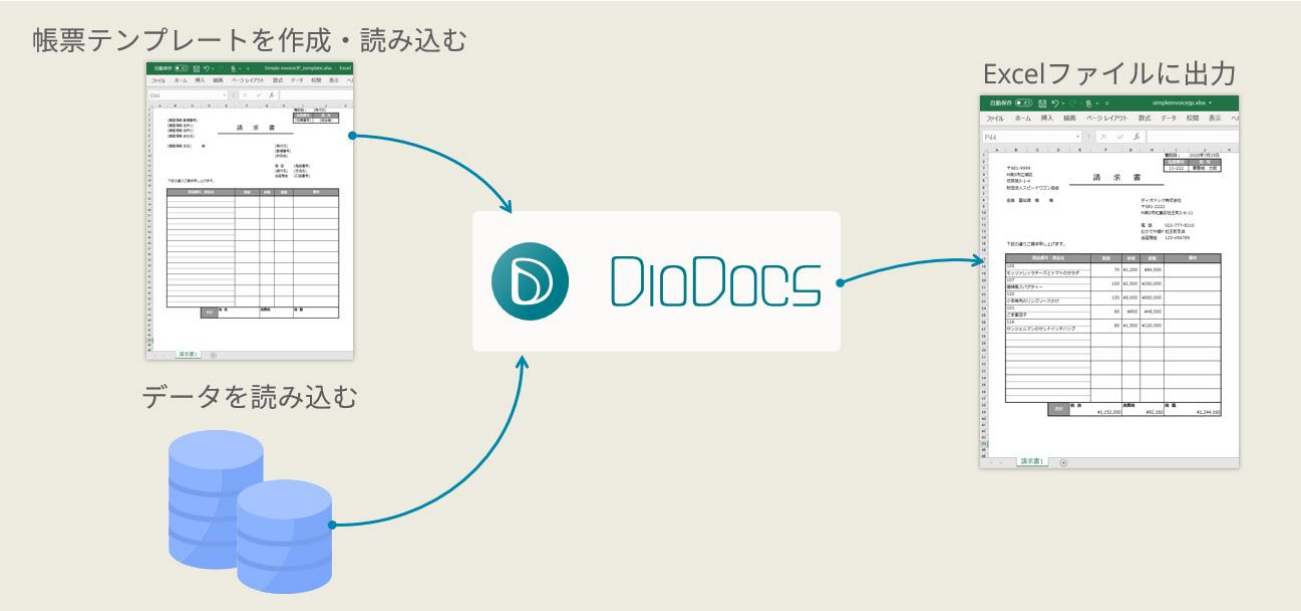


図 1　Excel 帳票の基本フロー

データの埋め込み（コード）

ワークシートの「A1」や「A2」といった個別のセル、あるいは「A1:A5」といったセル範囲に対して、ひとつずつコードを記載してデータをセットします。VBA や VSTO の標準的な手法と一緒なので学習コストが低く手軽である反面、帳票レイアウトの複雑さやデータ数に比例してコード量が膨大になるという欠点があります。したがって、単票のようなシンプルで静的な固定帳票におすすめの手法です。

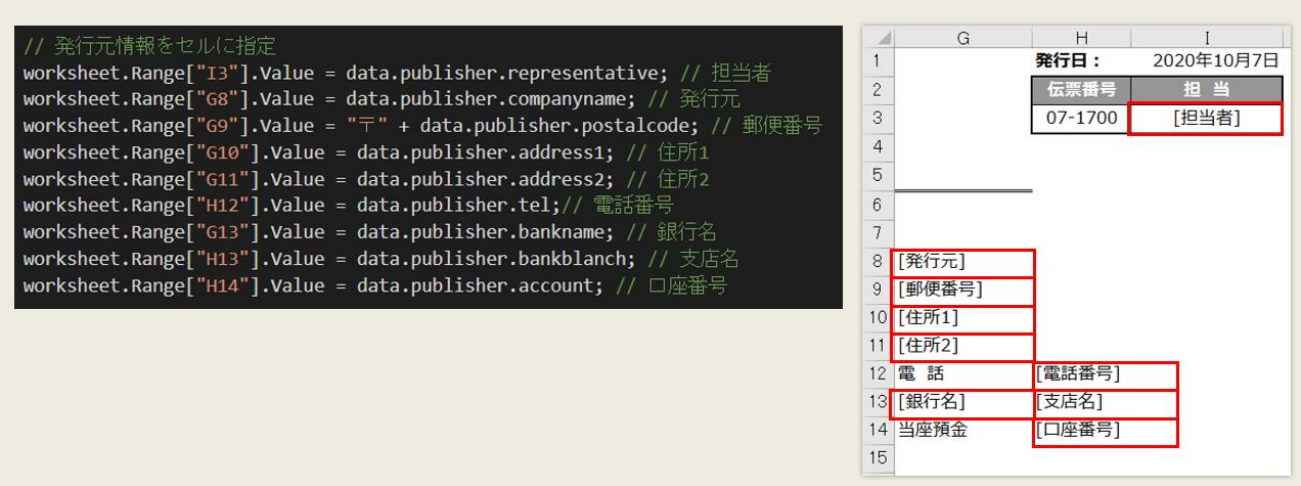


図 2　請求書（鏡）へのデータ埋め込み（コード）

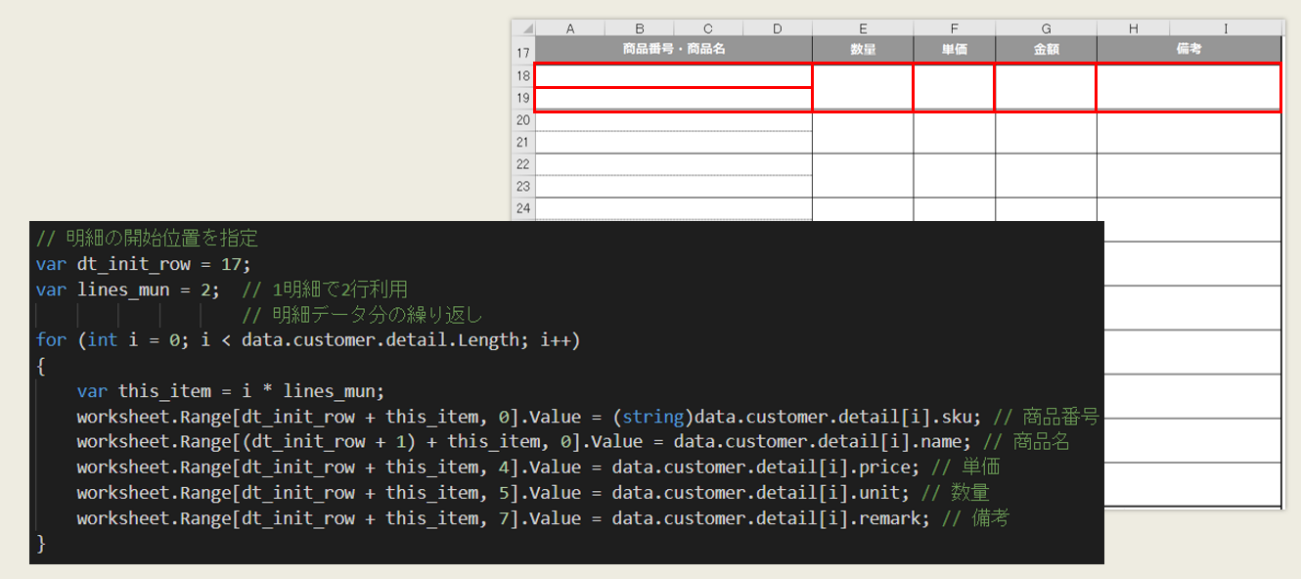


図 3　請求書（明細）へのデータ埋め込み（コード）

データの埋め込み（テンプレート構文）

DioDocs 独自のテンプレート構文を利用した方法です。ActiveReports のような帳票専用ツールと似たコンセプトと言うことができるでしょう。データを埋め込みたいセルに対してデータ連結フィールドを意味する独自のテンプレート構文を記述します。DioDocs はこの構文が記載されたセルを特別なセルとして認識し、帳票生成時にデータを自動的に埋め込み、必要に応じてレイアウトを変更します。この方法は「複数行の明細を出力したい」や「グループ毎に改ページしたい」などの動的な帳票を生成する場合におすすめの手法です。

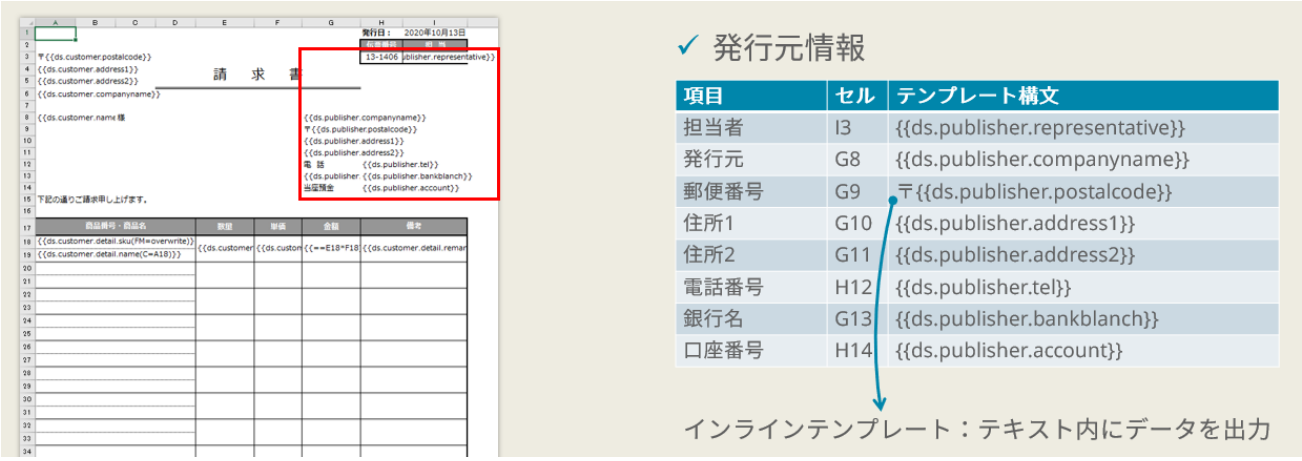


図 4　請求書（鏡）へのデータ埋め込み（テンプレート構文）

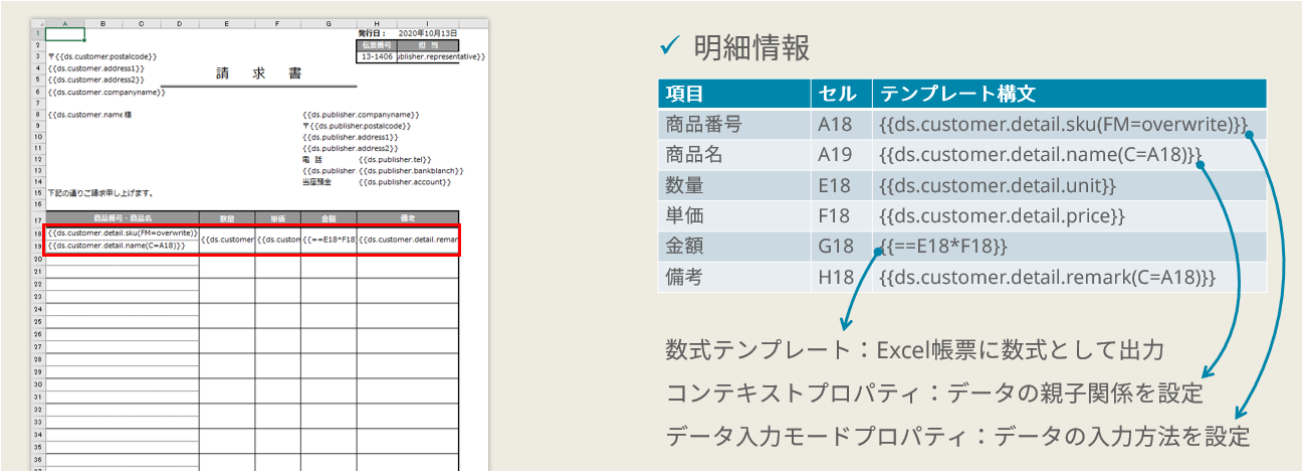


図 5　請求書明細へのデータ埋め込み（テンプレート構文）

テンプレート構文を利用して Excel 帳票を出力する際の基本的な手順は次の通りです。

(1)　テンプレートの読み込み

```
workbook.Open("Template.xlsx");
```

(2)　データの追加

```
workbook.AddDataSource("ds", datasource);
```

(3)　テンプレートの実行

```
workbook.ProcessTemplate();
```

(4)　Excel や PDF に保存

```
workbook.Save("report.xlsx");
workbook.Save("report.pdf ");
```

テンプレート構文の概要

テンプレート構文は二重の波カッコ「**{{hogehoge}}**」を使用して表現します。
図 6 は最もシンプルなテンプレート構文の例で、部署と氏名の 2 カラムで構成される表形式の帳票を作成するためのものです。部署列に**{{ds1.DevTeams.Team}}**が、氏名列には**{{ds1.DevTeams.Name}}**が記述されています。

	A	B	C	D
1				
2		部署	氏名	
3		{{ds1.DevTeams.Team}}	{{ds1.DevTeams.Name}}	
4				

図 6　テンプレート構文の例

このテンプレート構文を含んだ帳票テンプレートを DioDocs で読み込んで実行すると図 7 の結果が得られます。

	A	B	C	D
1				
2		部署	氏名	
3		仙台	桂三郎	
4			高森次郎	
5			寺岡花子	
6			葡萄太郎	
7		東京	御徒町久美子	
8			秋葉原孝太郎	
9			上野大輔	
10				

図 7　テンプレート構文の実行結果

最終的に 7 行のデータが展開され、部署列は同一データ「仙台」、「東京」を持つセルがマージされています。テンプレート構文を使わずに同じ結果を得るにはコードを複数行記載する必要があるでしょう。

テンプレート構文を使用した場合はコードを一切記述する必要がなく、レイアウトやグループ化の設定などはテンプレート内で完結します。このため、帳票を運用するフェーズでレイアウトの変更が必要になった場合、プログラムコードに手を加えることなく帳票テンプレートの変更のみで対応が可能になるというメリットがあります。

テンプレート構文の詳細

データソース

テンプレート構文を使って Excel 帳票へデータを出力する場合、まずはデータソースの定義が必要です。この定義は DioDocs を使用するアプリケーションのプログラムコードで定義します。

```
var reportdata1 = new TestData();
reportdata1.DevTeams = new List<DevTeam>
{
    new DevTeam { Team = "仙台", Name = "葡萄太郎" },
    new DevTeam { Team = "仙台", Name = "寺岡花子" },
    new DevTeam { Team = "仙台", Name = "高森次郎" },
    new DevTeam { Team = "東京", Name = "秋葉原孝太郎" },
    new DevTeam { Team = "東京", Name = "御徒町久美子" },
    new DevTeam { Team = "東京", Name = "上野大輔" },
    new DevTeam { Team = "仙台", Name = "桂三郎" }
};

// データソースをテンプレートに追加
reportbook1.AddDataSource("ds1", reportdata1);

// レポートの生成
reportbook1.ProcessTemplate();
```

図 8　データソースの準備（DioDocs のプログラムコード）

図 8 では、reportdata1 という TestData 型の変数を作成しデータを追加した後、AddDataSource メソッドを使って ds1 という名称でワークブックに追加しています。こうすることで Excel ワークブックからテンプレート構文を介して TestData 型のレポートデータにアクセスできるようになります。図 6 で紹介したテンプレート構文 **{{ds1.DevTeams.Team}}**は図 8 のデータソース定義があって初めて有効になる構文です。

DioDocs が対応しているデータソースはユーザー定義型のカスタムオブジェクト（デシリアライズした JSON オブジェクトも利用可能）、DataTable／DataSet、変数、配列／リストです。

データソースに関する参考情報

製品ヘルプ：

https://docs.grapecity.com/help/diodocs/excel/#data_source_binding.html

製品デモ：

https://demo.grapecity.com/diodocs/excel/datasource

テンプレートフィールド

{{～}}で囲まれた部分をテンプレートフィールド（またはテンプレートセル）と呼びます。テンプレートフィールドには以下の種類があります。

データフィールド

データソースのデータが入力される連結フィールドです。以下の 3 種類があります。

名前	帳票テンプレートの表示	テンプレート実行後の表示 （データはダミーです）
データフィールド	**{{ds.count}}**	5
入れ子データフィールド	**{{ds.task.dueDate}}**	12 月 4 日
インラインデータフィールド	**〒 {{ds.PostalCode }}**	〒 981-2222

関数フィールド

DioDocs でテンプレート構文を実行する際に計算を行うために使用します。セルやデータフィールドに適用することができます。テンプレート構文を実行した後の Excel 帳票上では数値データになります。式は維持されません。

帳票テンプレートの表示	テンプレート実行後の表示 （データはダミーです）	テンプレート実行後の数式 （式は維持されません）
{{=SUM(F4) }}	1000	1000
{{=Count(student.name}}	10	10

式フィールド

式フィールドは、演算子「+」、「-」、「*」、「/」、「()」を使用した計算を行うために使用します。セルやデータフィールドに適用することができます。関数フィールドと同様にテンプレート構文を実行した後の Excel 帳票上では数値データになります。式は維持されません。

帳票テンプレートの表示	テンプレート実行後の表示 （データはダミーです）	テンプレート実行後の数式 （式は維持されません）
{{=ds.count*ds.prise }}	1000	1000
{{=A18*0.05}}	50	50
{{=(A18+ A20)*0.3}}	300	300

数式フィールド

関数フィールド、式フィールドと似ていますが、「=」が 2 個ある点に注意してください。この「＝＝」のように記載するとテンプレート構文を実行した後で数式が維持されます。Excel 帳票で値だけでなく計算式も維持したい場合に利用します。

帳票テンプレートの表示	テンプレート実行後の表示 （データはダミーです）	テンプレート実行後の数式 （式は維持されます）
{{==**SUM(F4)** }}	1000	=SUM(F4)
{{==**A18*0.05**}}	50	=A18*0.05

シート名フィールド

ワークシートの名前にデータフィールドを利用できます。シート名のフィールド順にデータソースからデータが入力され複数のワークシートが作成されます。例えば、データソースが 5 つの地域のデータを含む場合、帳票テンプレートで**{{dt.Region}}**をシート名として指定すると、以下のように Excel 帳票では各地域のデータを含む 5 枚のワークシートが作成されます。

図 9　シート名フィールド

テンプレートフィールドに関する参考情報

製品ヘルプ：

https://docs.grapecity.com/help/diodocs/excel/#template-fields.html

製品デモ：

https://demo.grapecity.com/diodocs/excel/templatecell

テンプレートプロパティ

テンプレートフィールドとともに丸括弧「()」に囲まれて定義されます。プロパティはキーと値の形式で設定されます。 複数のプロパティはコンマを使用して区切ります。

例：{{**ds.field(G=list, C=A2)**}}

セルコンテキスト

データの親子関係を指定します。{{**ds.field(C=A1)**}}のように定義します。

1. デフォルトのコンテキスト

プロパティを何も指定しなかった場合、隣接する左側のセルが親、右側のセルが子となるよう暗黙的なコンテキストが設定されます（セル拡張が E=H と設定されている場合は左側でなく上側の隣接セルが親になります）。これは「商品名」データフィールドが「カテゴリ」データフィールドと親子関係、「{{**ds.Name(C=ds.Category)**}}」あるいは「{{**ds.Name(C=A2)**}}」の状態であると言えます。これをデフォルトのコンテキストと呼びます。

帳票テンプレートの表示

カテゴリ	商品名
{{ds.Category}}	{{ds.Name}}

テンプレート実行後の表示

カテゴリ	商品名
家電製品	Bose 785593-0050
	Canon EOS 1500D
	Haier 394L 4Star
	IFB 6.5 Kg FullyAuto
	Mi LED 40inch
	Sennheiser HD 4.40-BT
携帯電話	Iphone XR
	OnePlus 7Pro
	Redmi 7
	Samsung S9

2. コンテキストなし

「**C=None**」または「**C=N**」と記述するとデフォルトのコンテキストを削除できます。以下の帳票テンプレートの場合、カテゴリと商品名の親子関係が解除され、それぞれのデータが独立して表示されます。

帳票テンプレートの表示

カテゴリ	商品名
{{ds.Category}}	{{ds.Name(C=None)}}

テンプレート実行後の表示

カテゴリ	商品名
家電製品	Bose 785593-0050
携帯電話	Canon EOS 1500D
	Haier 394L 4Star
	IFB 6.5 Kg FullyAuto
	Iphone XR
	Mi LED 40inch
	OnePlus 7Pro
	Redmi 7
	Samsung S9
	Sennheiser HD 4.40-BT

3. カスタムコンテキスト

この例ではカテゴリと商品名のデータフィールドが隣接していないためデフォルトのコンテキストは有効になりません。商品名のデータフィールドにコンテキスト「**C=A2**」を記述することで親子関係を設定できます。

帳票テンプレートの表示

カテゴリ	商品名
{{ds.Category}}	
	{{ds.Name(C=A2)}}

テンプレート実行後の表示

カテゴリ	商品名
家電製品	
	Bose 785593-0050
	Canon EOS 1500D
	Haier 394L 4Star
	IFB 6.5 Kg FullyAuto
	Mi LED 40inch
	Sennheiser HD 4.40-BT
携帯電話	
	Iphone XR
	OnePlus 7Pro
	Redmi 7
	Samsung S9

セル拡張

データフィールドの展開方向を指定します。**{{ds.field(E=H)}}**のように定義します。

1. 縦方向への展開（デフォルト）

プロパティを何も指定しなかった場合や「**E=Vertical**」または「**E=V**」と指定した場合、データフィールドは縦方向に上から下へ展開されます。

帳票テンプレートの表示

カテゴリ	商品名
{{ds.Category}}	{{ds.Name}}

テンプレート実行後の表示

カテゴリ	商品名
家電製品	Bose 785593-0050
	Canon EOS 1500D
	Haier 394L 4Star
	IFB 6.5 Kg FullyAuto
	Mi LED 40inch
	Sennheiser HD 4.40-BT
携帯電話	Iphone XR
	OnePlus 7Pro
	Redmi 7
	Samsung S9

2. 横方向への展開

「**E=Horizontal**」または「**E=H**」を指定すると、横方向に左から右へデータが展開されます。

帳票テンプレートの表示

カテゴリ	{{ds.Category(E=H)}}
商品名	{{ds.Name(E=H)}}

テンプレート実行後の表示

カテゴリ	家電製品						携帯電話	
商品名	Bose 785593-0050	Canon EOS 1500D	Haier 394L 4Star	IFB 6.5 Kg FullyAuto	Mi LED 40inch	Sennheiser HD 4.40-E	Iphone XR	OnePlus 7Pro

グループ化

データフィールドのグループ化方法を指定します。**{{ds.field(G=Merge)}}**のように定義します。

1. マージ（デフォルト）

プロパティを何も指定しなかった場合や「**G=M**」または「**G=Merge**」を指定した場合、同一グループのセルがマージされます。

帳票テンプレートの表示

カテゴリ	商品名
{{ds.Category}}	{{ds.Name}}

テンプレート実行後の表示

カテゴリ	商品名
家電製品	Bose 785593-0050
	Canon EOS 1500D
	Haier 394L 4Star
	IFB 6.5 Kg FullyAuto
	Mi LED 40inch
	Sennheiser HD 4.40-BT
携帯電話	Iphone XR
	OnePlus 7Pro
	Redmi 7
	Samsung S9

2. ノーマル

「**G=N**」または「**G＝Normal**」を指定した場合、グループ化されますがセルのマージは行われません。

帳票テンプレートの表示

カテゴリ	商品名
{{ds.Category(G=N)}}	{{ds.Name}}

テンプレート実行後の表示

カテゴリ	商品名
家電製品	Bose 785593-0050
	Canon EOS 1500D
	Haier 394L 4Star
	IFB 6.5 Kg FullyAuto
	Mi LED 40inch
	Sennheiser HD 4.40-BT
携帯電話	Iphone XR
	OnePlus 7Pro
	Redmi 7
	Samsung S9

3. リピート

「**G=R**」または「**G=Repeat**」を指定した場合、グループ化されグループのキー値が繰り返し表示されます。

帳票テンプレートの表示

カテゴリ	商品名
{{ds.Category(G=R)}}	{{ds.Name}}

テンプレート実行後の表示

カテゴリ	商品名
家電製品	Bose 785593-0050
家電製品	Canon EOS 1500D
家電製品	Haier 394L 4Star
家電製品	IFB 6.5 Kg FullyAuto
家電製品	Mi LED 40inch
家電製品	Sennheiser HD 4.40-B1
携帯電話	Iphone XR
携帯電話	OnePlus 7Pro
携帯電話	Redmi 7
携帯電話	Samsung S9

4. リスト

「**G=L**」または「**G=List**」はデータソースの全てのレコードが展開される特別なオプションです。
帳票テンプレート上のデータフィールドはカテゴリと商品名の 2 つですが、データソースはこれらに地域、都市、売上の 3 つを加えた計 5 つのフィールドで構成され、全部で 26 レコード存在します。
したがって、テンプレート実行後は 26 レコード全てが表示されています。

帳票テンプレートの表示	テンプレート実行後の表示

カテゴリ	商品名
{{ds.Category(G=L)}}	{{ds.Name}}

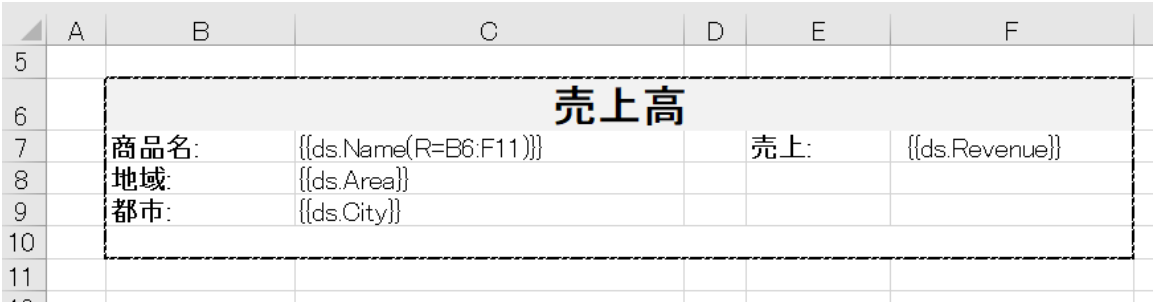
カテゴリ	商品名
家電製品	Bose 785593-0050
家電製品	Bose 785593-0050
家電製品	Bose 785593-0050
家電製品	Canon EOS 1500D
家電製品	Canon EOS 1500D
家電製品	Canon EOS 1500D
家電製品	Haier 394L 4Star
家電製品	Haier 394L 4Star
家電製品	Haier 394L 4Star
家電製品	IFB 6.5 Kg FullyAuto
家電製品	IFB 6.5 Kg FullyAuto
家電製品	IFB 6.5 Kg FullyAuto
家電製品	Mi LED 40inch
家電製品	Mi LED 40inch
家電製品	Mi LED 40inch
家電製品	Sennheiser HD 4.40-BT
家電製品	Sennheiser HD 4.40-BT
携帯電話	Iphone XR
携帯電話	Iphone XR
携帯電話	OnePlus 7Pro
携帯電話	OnePlus 7Pro
携帯電話	Redmi 7
携帯電話	Redmi 7
携帯電話	Samsung S9
携帯電話	Samsung S9
携帯電話	Samsung S9

セル範囲

複数のデータフィールドからなる範囲をグループ化する場合に使用します。**{{ds.field(R=A1:C3)}}**のように定義します。これは Access や ActiveReports といったレポートツールにおけるセクションという概念に似ています。セル範囲を指定すると、その範囲内に含まれるデフォルトや明示的なコンテキストのないデータフィールドは、セル範囲が定義されたデータフィールドをセルコンテキストと解釈します。

以下の Excel テンプレートでは商品名データフィールドに対してセル範囲「**(R=B6:F11)**」を定義し、商品に紐づく詳細情報をグループ化しています。このテンプレートを実行すると **B6:F11** の範囲が商品の数だけ繰り返し表示されます。

帳票テンプレートの表示



テンプレート実行後の表示

	A	B	C	D	E	F
5						
6		売上高				
7		商品名:	Bose 785593-0050		売上:	¥ 92,800.00
8		地域:	北米			
9		都市:	シカゴ			
10						
11						
12		売上高				
13		商品名:	Canon EOS 1500D		売上:	¥ 98,650.00
14		地域:	北米			
15		都市:	シカゴ			
16						
17						
18		売上高				
19		商品名:	Haier 394L 4Star		売上:	¥ 367,050.00
20		地域:	北米			
21		都市:	シカゴ			
22						
23						
24		売上高				
25		商品名:	IFB 6.5 Kg FullyAuto		売上:	¥ 673,800.00
26		地域:	南米			
27		都市:	ブエノスアイレス			
28						
29						
30		売上高				
31		商品名:	Iphone XR		売上:	¥ 1,734,621.00
32		地域:	北米			
33		都市:	ミネソタ			
34						
35						

ソート

データフィールドのソート方法を指定します。{{**ds.field**(**S=Desc**)}}のように定義します。

1. 昇順（デフォルト）

プロパティを何も指定しなかった場合や「**S=A**」または「**S=Asc**」を指定した場合、昇順にソートされます。

帳票テンプレートの表示	テンプレート実行後の表示
商品名 {{ds.Name}}	**商品名** Bose 785593-0050 Canon EOS 1500D Haier 394L 4Star IFB 6.5 Kg FullyAuto Iphone XR Mi LED 40inch OnePlus 7Pro Redmi 7 Samsung S9 Sennheiser HD 4.40-BT

2. 降順

「**S=D**」または「**S=Desc**」を指定した場合、降順にソートされます。

<table>
<tr><th>帳票テンプレートの表示</th><th>テンプレート実行後の表示</th></tr>
<tr><td>
<table>
<tr><th>商品名</th></tr>
<tr><td>{{ds.Name(S=desc)}}</td></tr>
</table>
</td><td>
<table>
<tr><th>商品名</th></tr>
<tr><td>Sennheiser HD 4.40-BT</td></tr>
<tr><td>Samsung S9</td></tr>
<tr><td>Redmi 7</td></tr>
<tr><td>OnePlus 7Pro</td></tr>
<tr><td>Mi LED 40inch</td></tr>
<tr><td>Iphone XR</td></tr>
<tr><td>IFB 6.5 Kg FullyAuto</td></tr>
<tr><td>Haier 394L 4Star</td></tr>
<tr><td>Canon EOS 1500D</td></tr>
<tr><td>Bose 785593-0050</td></tr>
</table>
</td></tr>
</table>

3. なし

「**S=N**」または「**S=None**」を指定した場合、ソートされません。

Excel テンプレートの表示	テンプレート実行後の表示
商品名 {{ds.Name(S=none)}}	商品名 Bose 785593-0050 Canon EOS 1500D Haier 394L 4Star IFB 6.5 Kg FullyAuto Mi LED 40inch Sennheiser HD 4.40-BT Iphone XR OnePlus 7Pro Redmi 7 Samsung S9

ページ区切り

データフィールドの後にページ区切りを追加するかどうかを指定します。{{**ds.field(PB=True)**}}または{{**ds.field(Pagebreak=True)**}}のように定義します。

1. 水平ページ区切り

先頭列にページ区切りが指定されている場合、水平ページ区切りが追加されます。

帳票テンプレートの表示

	A	B
1	カテゴリ	商品名
2	{{ds.Category(Pb=true)}}	{{ds.Name}}

テンプレート実行後の表示（改ページプレビュー）

	A	B
1	カテゴリ	商品名
2	家電製品 1 ページ	Bose 785593-0050
3		Canon EOS 1500D
4		Haier 394L 4Star
5		IFB 6.5 Kg FullyAuto
6		Mi LED 40inch
7		Sennheiser HD 4.40-BT
8	携帯電話 2 ページ	Iphone XR
9		OnePlus 7Pro
10		Redmi 7
11		Samsung S9

2. 垂直ページ区切り

先頭行にページ区切りが指定されている場合、垂直ページ区切りが追加されます。

帳票テンプレートの表示

	A	B
1		カテゴリ{{(PB=True)}}
2		{{ds.Category(E=H, R=B1:B4)}}
3		商品名
4		{{ds.Name}}

テンプレート実行後の表示（改ページプレビュー）

	A	B	C
1		カテゴリ	カテゴリ
2		家電製品	携帯電話
3		商品名	商品名
4		Bose 785593-0050	Iphone XR
5		Canon EOS 1500D	OnePlus 7Pro
6		Haier 394L 4Star	Redmi 7
7		IFB 6.5 Kg FullyAuto	Samsung S9
8		Mi LED 40inch	
9		Sennheiser HD 4.40-BT	

1ページ　2ページ

塗りつぶしモード

データフィールドの展開方法を指定します。{{**ds.field(FM=Insert)**}}のように定義します。

1. 挿入

プロパティを何も指定しなかった場合、もしくは「**FM=Insert**」を指定した場合、空白行を挿入しその行にテンプレートのスタイルをセットしデータを出力します。データソースのレコード数分だけ行が挿入されます。

帳票テンプレートの表示

	A	B	C	D
5				
6	四半期の業績	第4四半期 売上高		
7				
8				
9	事業名:	Eコマース		
10				
11	**地域**	**都市**	**商品名**	**売上**
12	{{ds.Area(G=list)}}	{{ds.City}}	{{ds.Name}}	{{ds.Revenue}}
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24	**地域別売上高**			**$0.00**

テンプレート実行後の表示

	A	B	C	D
5				
6	四半期の業績	第4四半期 売上高		
7				
8				
9	事業名:	Eコマース		
10				
11	**地域**	**都市**	**商品名**	**売上**
12	北米	シカゴ	Bose 785593-0050	$92,800.00
13	北米	シカゴ	Haier 394L 4Star	$367,050.00
14	南米	サンティアゴ	Haier 394L 4Star	$578,900.00
15	南米	メデジン	IFB 6.5 Kg FullyAuto	$82,910.00
16	北米	シカゴ	Sennheiser HD 4.40-BT	$178,100.00
17	南米	キト	OnePlus 7Pro	$215,000.00
18	北米	ミネソタ	Redmi 7	$81,650.00
19	南米	キト	Samsung S9	$716,520.00
20				
21				
22				
23				
24				
25				
26				
27				
28				
29				
30				
31	**地域別売上高**			**$2,312,930.00**

2. 上書き

「**FM=Overwrite**」を指定した場合、テンプレート上に空白行は挿入されずに、現在あるセル上にデータフィールドが展開されます。これは帳票の明細数が決まっている場合など、固定レイアウトの帳票に有効なオプションです。

帳票テンプレートの表示

	A	B	C	D
5				
6	四半期の業績	第4四半期 売上高		
7				
8				
9	事業名:	Eコマース		
10				
11	**地域**	**都市**	**商品名**	**売上**
12	{{ds.Area(G=list, FM=overwrite)}}	{{ds.City}}	{{ds.Name}}	{{ds.Revenue}}
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24	**地域別売上高**			**$0.00**

テンプレート実行後の表示

	A	B	C	D
5				
6	四半期の業績	第4四半期 売上高		
7				
8				
9	事業名:	Eコマース		
10				
11	**地域**	**都市**	**商品名**	**売上**
12	北米	シカゴ	Bose 785593-0050	$92,800.00
13	北米	シカゴ	Haier 394L 4Star	$367,050.00
14	南米	サンティアゴ	Haier 394L 4Star	$578,900.00
15	南米	メデジン	IFB 6.5 Kg FullyAuto	$82,910.00
16	北米	シカゴ	Sennheiser HD 4.40-BT	$178,100.00
17	南米	キト	OnePlus 7Pro	$215,000.00
18	北米	ミネソタ	Redmi 7	$81,650.00
19	南米	キト	Samsung S9	$716,520.00
20				
21				
22				
23				
24	**地域別売上高**			**$1,121,660.00**

3. 上書き ＋ データのオーバーフロー

テンプレートに定義された明細行数以上のデータを出力した場合、はみ出したデータによってレイアウトが崩れてしまいます。**fillRange**（**FR**）プロパティを使用すると明細行を自動拡大することができます。
以下の例では、テンプレートに定義された明細行数は 12 行です。**fillRange** プロパティには A12:D23 の 12 行分のセル範囲を指定しています。この状況で 16 行のデータとバインドすると、明細行が全部で 24 行（元々の 12

行と追加の 12 行）に拡大されて全データを表示できるようになります。

Excel テンプレートの表示

	A	B	C	D
5				
6	四半期の業績	第4四半期 売上高		
7				
8				
9	事業名:	Eコマース		
10				
11	地域	都市	商品名	売上
12	{{ds.Area(G=list, FM=overwrite, FR=A12:D23)}}		{{ds.Name}}	{{ds.Revenue}}
13				
14				
15				
16				
17				
18				
19				
20				
21				
22				
23				
24	地域別売上高			$0.00

テンプレート実行後の表示

	A	B	C	D
5				
6	四半期の業績	第4四半期 売上高		
7				
8				
9	事業名:	Eコマース		
10				
11	地域	都市	商品名	売上
12	北米	シカゴ	Bose 785593-0050	$92,800.00
13	北米	シカゴ	Haier 394L 4Star	$367,050.00
14	南米	サンティアゴ	Haier 394L 4Star	$578,900.00
15	南米	メデジン	IFB 6.5 Kg FullyAuto	$82,910.00
16	北米	シカゴ	Sennheiser HD 4.40-BT	$178,100.00
17	南米	キト	OnePlus 7Pro	$215,000.00
18	北米	ミネソタ	Redmi 7	$81,650.00
19	南米	キト	Samsung S9	$716,520.00
20	北米	シカゴ	Bose 785593-0050	$92,800.00
21	北米	シカゴ	Haier 394L 4Star	$367,050.00
22	南米	サンティアゴ	Haier 394L 4Star	$578,900.00
23	南米	メデジン	IFB 6.5 Kg FullyAuto	$82,910.00
24	北米	シカゴ	Sennheiser HD 4.40-BT	$178,100.00
25	南米	キト	OnePlus 7Pro	$215,000.00
26	北米	ミネソタ	Redmi 7	$81,650.00
27	南米	キト	Samsung S9	$716,520.00
28				
29				
30				
31				
32				
33				
34				
35				
36	地域別売上高			$1,121,660.00

テンプレートプロパティに関する参考情報

製品ヘルプ：

https://docs.grapecity.com/help/diodocs/excel/#template_properties.html

製品デモ：

https://demo.grapecity.com/diodocs/excel/properties

請求書の作成

ここからはテンプレート構文を使って実際に請求書を作成する手順を紹介します。DioDocs を利用すると Excel 帳票のみならず PDF 帳票も簡単に作成することができます。

開発環境の準備

開発ツールは Visual Studio Code と .NET 6 の SDK を利用します。それぞれ以下のリンク先から取得してください。これらはクロスプラットフォームで動作しますが、本資料では Windows 環境を使用します。

Visual Studio Code：
https://code.visualstudio.com/download

.NET 6 SDK：
https://dotnet.microsoft.com/ja-jp/download

この SDK に含まれている .NET CLI を利用してコマンドラインからプロジェクトの作成や実行を行います。 .NET CLI の詳細は以下のドキュメントを参照してください。

.NET CLI：
https://docs.microsoft.com/ja-jp/dotnet/core/tools/

帳票テンプレート（*.xlsx）の作成と編集には Microsoft Excel を利用します。
https://www.microsoft.com/ja-jp/microsoft-365/excel

帳票出力（Excel ファイルおよび PDF ファイルへの出力）で利用するフォントは「IPA P 明朝フォント」を利用します。文字情報技術促進協議会の Web サイトからダウンロードしてシステムにインストールしてください。
https://moji.or.jp/ipafont/ipa00303/

出力結果

作成した請求書の最終的な出力結果は下記のとおりです。

請求書（Excel ファイル出力）

１請求先が１シートとなるよう構成されています。また請求先毎に明細件数が異なります。

発行日： 2022年1月11日

伝票番号	担 当
11-1605	葡萄 城太郎

〒931-9999
M県S市広瀬区
花京院3-1-4
財団法人スピ--ドワゴン

請 求 書

虎猿出井 富似雄 様

ディオドック株式会社
〒931-2222
M県S市紅葉区
杜王町2-6-11
電 話 022-777-8210
むかでや銀行 杜王町支店
当座預金 123-456789

下記の通りご請求申し上げます。

商品番号･商品名	数量	単価	金額	備考
0105231 モッツァレッラチ--ズとトマトのサラダ	70	¥1,200	¥84,000	前菜
0201071 娼婦風スパゲティ--	100	¥2,500	¥250,000	パスタ
0301201 小羊背肉のリンゴソ--スかけ	130	¥5,000	¥650,000	メイン
0401011 ごま蜜団子	60	¥800	¥48,000	デザ--ト

合計	税 抜	消費税	総 額
	¥1,032,000	¥103,200	¥1,135,200

図 10　請求書（1 ページ目）

発行日： 2022年1月11日

伝票番号	担当
11-1605	葡萄 城太郎

〒982-1234
M県S市秋保区
上愛子6-4-7
杜王グランドホテル

請 求 書

江陽 太郎 様

ディオドック株式会社
〒981-2222
M県S市紅葉区
杜王町2-6-11
電 話 022-777-3210
むかでや銀行 杜王町支店
当座預金 123-456789

下記の通りご請求申し上げます。

商品番号・商品名	数量	単価	金額	備考
0202061 激辛麻婆焼きそば	50	¥1,000	¥50,000	麺
0302081 塩竈マグロ	30	¥2,300	¥184,000	魚
0402301 超極厚牛タン	110	¥4,800	¥528,000	肉
0502021 濃厚ずんだロール	60	¥800	¥48,000	デザート
0602271 ハチミツパンケーキ	60	¥1,300	¥78,000	お持ち帰り

合計	税抜	消費税	総額
	¥888,000	¥88,800	¥976,800

図 11　請求書（2 ページ目）

発行日： 2022年1月11日

伝票番号	担当
11-1605	葡萄 城太郎

〒983-5678
M県S市荒井区
深沼9-7-1
東方フルーツパーラー

請 求 書

東方 花子 様

ディオドック株式会社
〒981-2222
M県S市紅葉区
杜王町2-6-11
電 話 022-777-3210
むかでや銀行 杜王町支店
当座預金 123-456789

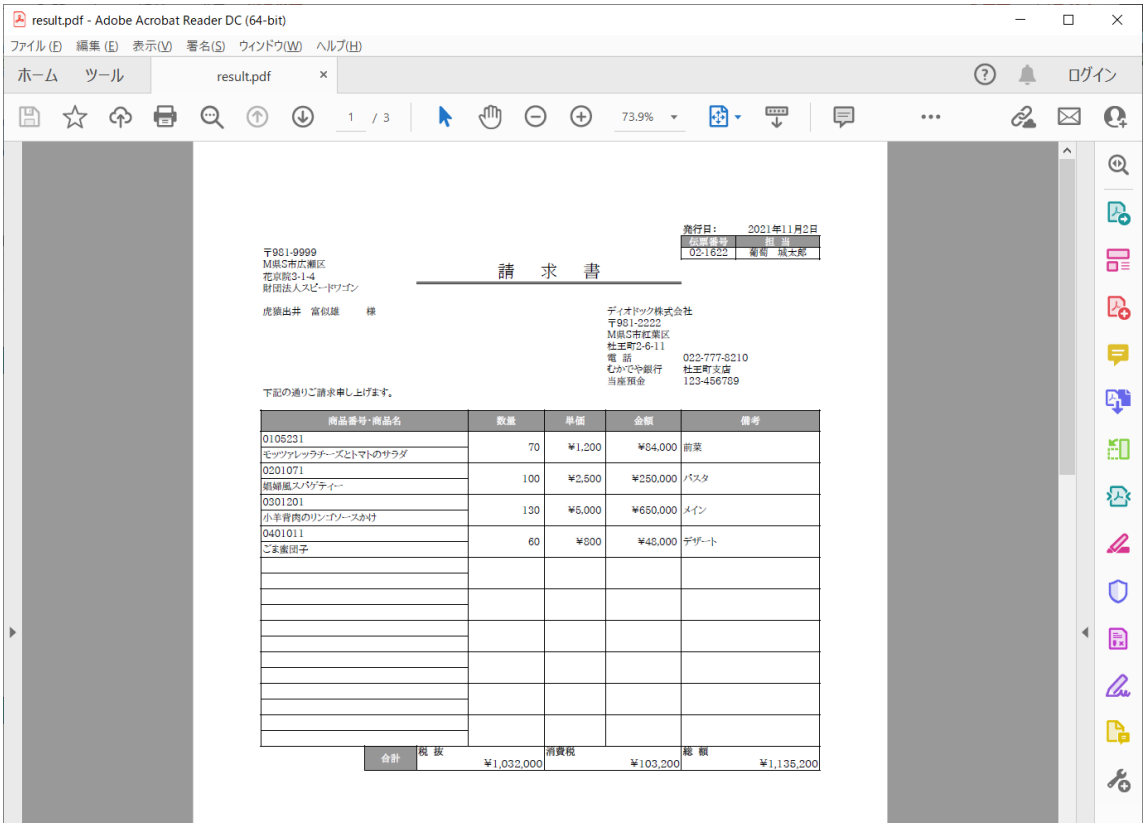
下記の通りご請求申し上げます。

商品番号・商品名	数量	単価	金額	備考
0303071 マスクメロン	60	¥1,100	¥66,000	フルーツ
0403091 シャインマスカット	90	¥2,400	¥216,000	フルーツ
0501161 サンジェルマンのサンドイッチバッグ	80	¥1,500	¥120,000	お持ち帰り
0503401 ピオーネ	120	¥4,900	¥588,000	フルーツ
0603031 ラ・フランス	50	¥700	¥35,000	フルーツ
0703381 富有柿	70	¥1,400	¥98,000	フルーツ

合計	税抜	消費税	総額
	¥1,123,000	¥112,300	¥1,235,300

図 12　請求書（3 ページ目）

請求書（PDF ファイル出力）



発行日：　2021年11月2日

伝票番号	担当
02-1622	葡萄　城太郎

〒981-9999
M県S市広瀬区
花京院3-1-4
財団法人スピードワゴン

虎猿出井　富似雄　様

請　求　書

ディオドック株式会社
〒981-2222
M県S市紅葉区
牡王町2-6-11
電　話　022-777-8210
むかでや銀行　牡王町支店
当座預金　123-456789

下記の通りご請求申し上げます。

商品番号・商品名	数量	単価	金額	備考
0105231 モッツァレッラチーズとトマトのサラダ	70	¥1,200	¥84,000	前菜
0201071 磯舗風スパゲティー	100	¥2,500	¥250,000	パスタ
0301201 小羊骨肉のリンゴソースかけ	130	¥5,000	¥650,000	メイン
0401011 ごま饅団子	60	¥800	¥48,000	デザート
合計	税抜 ¥1,032,000	消費税 ¥103,200	総額 ¥1,135,200	

図 13　請求書（1～3 ページ）

作成手順

下記の手順で作成していきます。完成済みのプロジェクトは GitHub で公開しています。

1. コンソールアプリケーションの作成

Invoice という空のフォルダを作成し Visual Studio Code でそのフォルダを開きます。ターミナル(PowerShell)を表示して **dotnet new console** とコマンド入力してコンソールアプリケーション新規作成します。

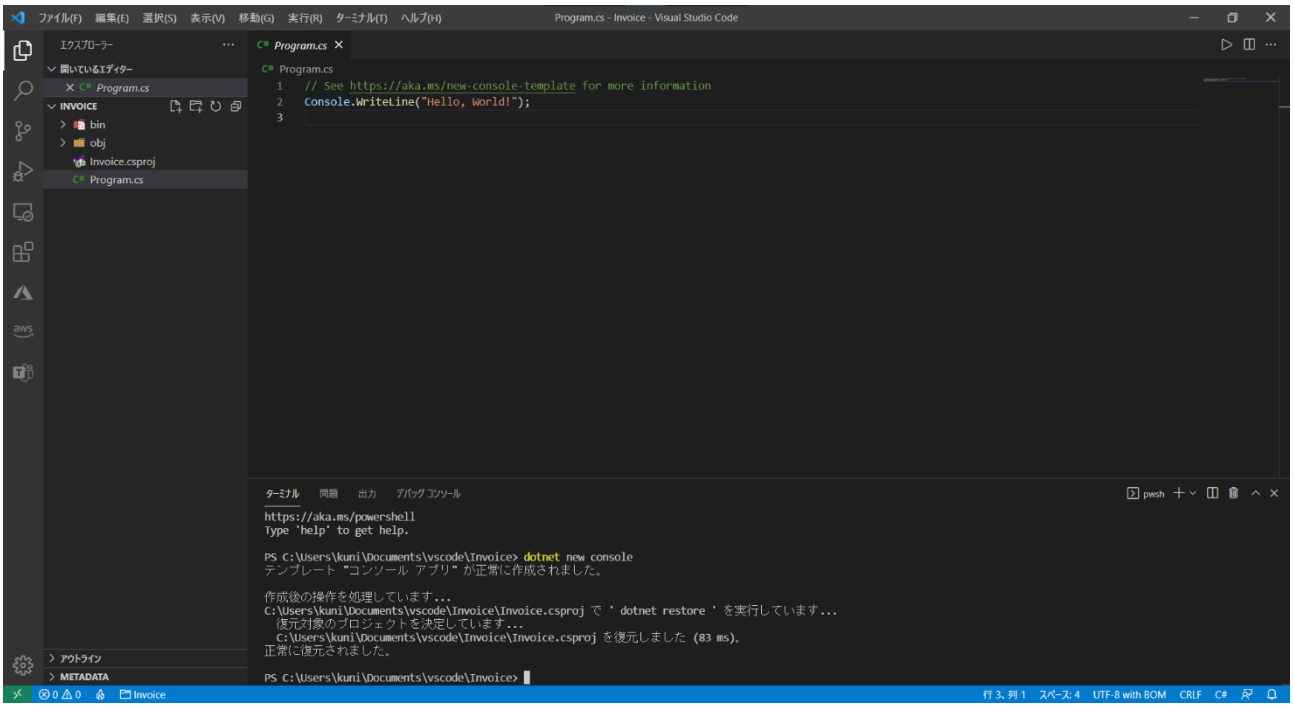


図 14　コンソールアプリケーションの新規作成

作成後に **dotnet run** とコマンド入力して動作を確認しておきます。「Hello, World!」がコンソールに表示されていればプロジェクトは正常に動作しています。次のステップに進んでください。

2. DioDocs のパッケージを追加

NuGet から DioDocs for Excel のパッケージを取得します。ターミナルで以下のコマンドを入力します。
dotnet add package GrapeCity.DioDocs.Excel.ja --version 4.2.5

成功すると、図 15 のようにプロジェクトファイル **Invoice.csproj** に **PackageReference** 要素が追加されます。

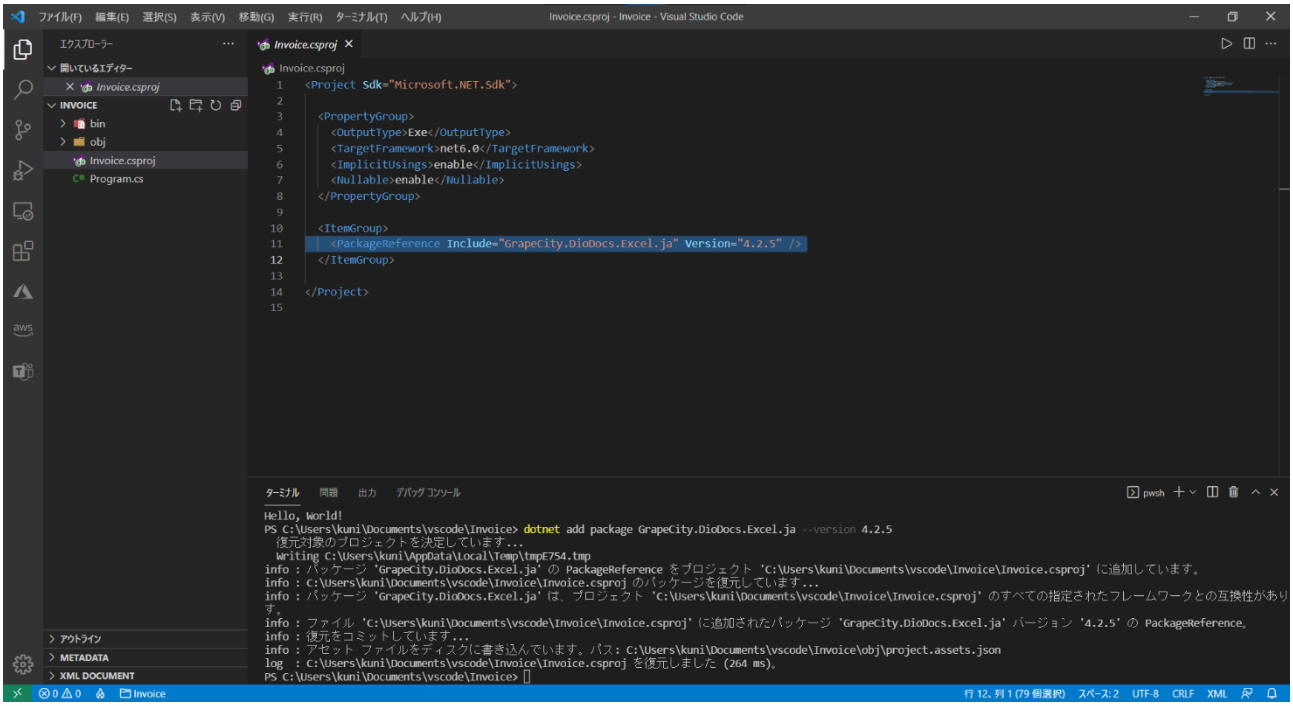


図 15　DioDocs のパッケージを追加

3. データソースの準備

データソースはカスタムオブジェクト（ユーザー定義オブジェクト）を利用します。以下の手順で作成します。

DataClass.cs の追加

プロジェクトに **DataClass.cs** という空のファイルを追加し、以下のコードをペーストします。

ファイル：DataClass.cs

```
public class Data
{
    public Publisher? publisher { get; set; }
    public Customer[]? customer { get; set; }
}

// 発行元情報
public class Publisher
{
    public string? companyname { get; set; }
    public string? postalcode { get; set; }
    public string? address1 { get; set; }
    public string? address2 { get; set; }
    public string? tel { get; set; }
    public string? bankname { get; set; }
    public string? bankblanch { get; set; }
```

```
    public string? account { get; set; }
    public string? representative { get; set; }
}

// 顧客情報
public class Customer
{
    public string? companyname{ get; set; }
    public string? name { get; set; }
    public string? postalcode { get; set; }
    public string? address1 { get; set; }
    public string? address2 { get; set; }
    public string? tel { get; set; }
    public Detail[]? detail { get; set; }
}

// 明細
public class Detail
{
    public string? sku { get; set; }
    public string? name { get; set; }
    public int price { get; set; }
    public int unit { get; set; }
    public string? remark { get; set; }
}
```

テストデータの作成

DataClass.cs で定義したカスタムオブジェクトからテストデータを作成します。**Program.cs** に以下のコードをペーストします。

ファイル：Program.cs

```
var data = new Data
{
    publisher = new Publisher
    {
        companyname = "ディオドック株式会社",
        postalcode = "981-2222",
        address1 = "M 県 S 市紅葉区",
        address2 = "杜王町 2-6-11",
        tel = "022-777-8210",
        bankname = "むかでや銀行",
        bankblanch = "杜王町支店",
        account = "123-456789",
        representative = "葡萄　城太郎"
    },
    customer = new Customer[]
    {
        new Customer
        {
            companyname = "財団法人スピードワゴン",
            name = "虎猿出井　富似雄",
            postalcode = "981-9999",
            address1 = "M 県 S 市広瀬区",
            address2 = "花京院 3-1-4",
            tel = "022-987-2220",
            detail = new Detail[]
            {
                 new Detail{ sku = "0105231", name = "モッツァレッラチーズとトマトのサラダ", price = 1200, unit =
70, remark = "前菜" },
                new Detail{ sku = "0201071", name = "娼婦風スパゲティー", price = 2500, unit = 100, remark =
"パスタ" },
                 new Detail{ sku = "0301201", name = "小羊背肉のリンゴソースかけ", price = 5000, unit = 130,
remark = "メイン" },
                new Detail{ sku = "0401011", name = "ごま蜜団子", price = 800, unit = 60, remark = "デザート
" }
            }
        },
```

```
        new Customer
        {
            companyname = "杜王グランドホテル",
            name = "江陽 太郎",
            postalcode = "982-1234",
            address1 = "M 県 S 市秋保区",
            address2 = "上愛子 6-4-7",
            tel = "022-876-1119",
            detail = new Detail[]
            {
                new Detail{ sku = "0202061", name = "激辛麻婆焼きそば", price = 1000, unit = 50, remark = "
麺" },
                new Detail{ sku = "0302081", name = "塩竈マグロ", price = 2300, unit = 80, remark = "魚" },
                new Detail{ sku = "0402301", name = "超極厚牛タン", price = 4800, unit = 110, remark = "肉
" },
                new Detail{ sku = "0502021", name = "濃厚ずんだロール", price = 800, unit = 60, remark = "デ
ザート" },
                new Detail{ sku = "0602271", name = "ハチミツパンケーキ", price = 1300, unit = 60, remark = "
お持ち帰り" }
            }
        },
        new Customer
        {
            companyname = "東方フルーツパーラー",
            name = "東方 花子",
            postalcode = "983-5678",
            address1 = "M 県 S 市荒井区",
            address2 = "深沼 9-7-1",
            tel = "022-765-0008",
            detail = new Detail[]
            {
                new Detail{ sku = "0303071", name = "マスクメロン", price = 1100, unit = 60, remark = "フルーツ
" },
                new Detail{ sku = "0403091", name = "シャインマスカット", price = 2400, unit = 90, remark = "フ
ルーツ" },
                new Detail{ sku = "0503401", name = "ピオーネ", price = 4900, unit = 120, remark = "フルーツ
" },
                new Detail{ sku = "0603031", name = "ラ･フランス", price = 700, unit = 50, remark = "フルーツ
" },
                new Detail{ sku = "0703381", name = "富有柿", price = 1400, unit = 70, remark = "フルーツ" },
                new Detail{ sku = "0501161", name = "サンジェルマンのサンドイッチバッグ", price = 1500, unit = 80,
remark = "お持ち帰り" }
```

```
            }
        }
    }
};
```

4. データソースの追加

DioDocs の API で新規ワークブックを作成し、そのワークブックに「3. データソースの準備」で作成したテストデータを追加します。**Program.cs** に以下のコードをペーストします。using ステートメントは先頭行にペーストしてください。

注）帳票テンプレート（請求書テンプレート.xlsx）はここではまだ存在しません。次の手順で作成します。

ファイル：Program.cs

```
using GrapeCity.Documents.Excel;

// 新しいワークブックを生成
var workbook = new Workbook();

// テンプレートを読み込む
workbook.Open("Templates/請求書テンプレート.xlsx");

// データソースを追加
workbook.AddDataSource("ds", data);
```

5. 帳票テンプレートの作成

プロジェクトのルートに Templates フォルダを作成し、その下に「請求書テンプレート.xlsx」を作成します。請求書のレイアウトを整え、データを表示したい箇所にテンプレート構文を記載していきます。

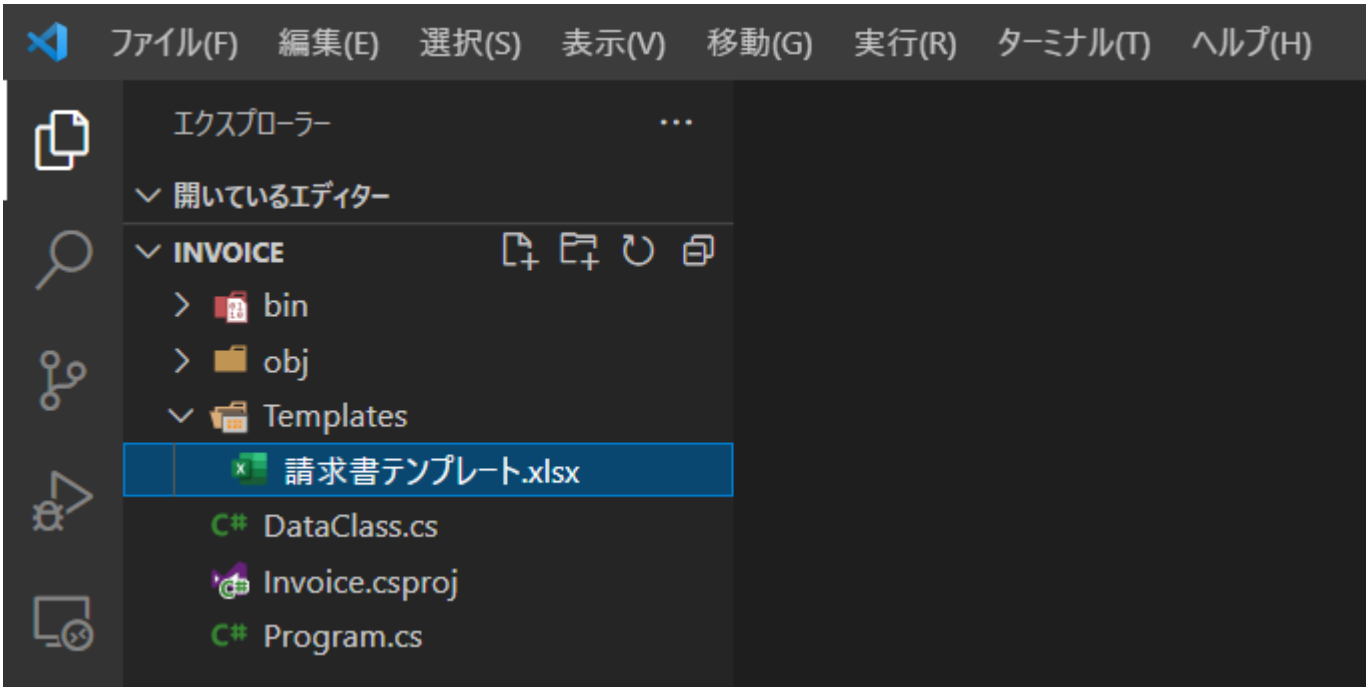


図 16　帳票テンプレートを追加

請求先情報

請求先となる個客情報を表示する部分を作成します。データソースの **Customer** クラスで定義されるフィールドをテンプレート構文で記載します。

表示項目	セル位置	テンプレート構文	フィールド種別
請求先郵便番号	A3	〒{{ds.customer.postalcode}}	インラインデータフィールド
請求先住所 1	A4	{{ds.customer.address1}}	データフィールド
請求先住所 2	A5	{{ds.customer.address2}}	同上
請求先企業名	A6	{{ds.customer.companyname}}	同上
請求先担当者名	A8	{{ds.customer.name}}	同上

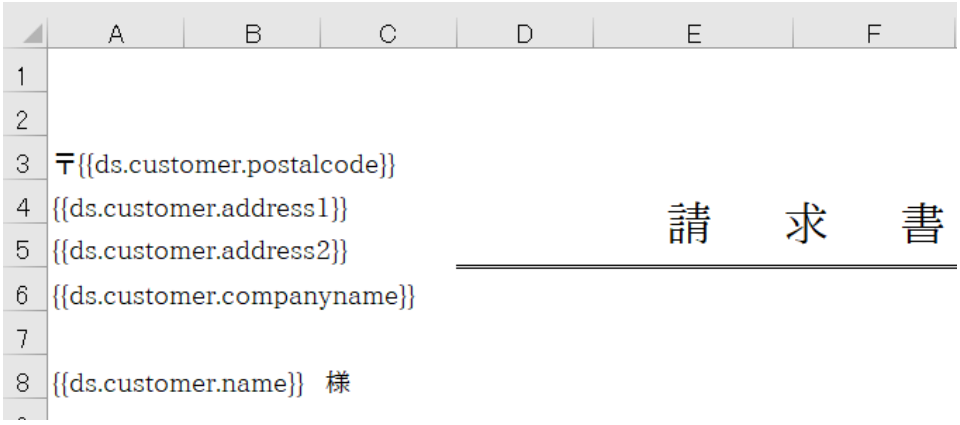


図 17　請求先情報

表示項目	セル位置	テンプレート構文	フィールド種別
請求先企業名	シートタブ	{{ds.customer.companyname}}	シート名フィールド

図 18　請求先情報（シート名）

請求元情報

請求元となる顧客情報を表示する部分を作成します。データソースの **Publisher** クラスで定義されるフィールドをテンプレート構文で記載します。

表示項目	セル位置	テンプレート構文	フィールド種別
請求元担当者	I3	{{ds.publisher.representative}}	データフィールド
請求元企業名	G8	{{ds.publisher.companyname}}	同上
請求元郵便番号	G9	〒{{ds.publisher.postalcode}}	インラインデータフィールド
請求元住所 1	G10	{{ds.publisher.address1}}	データフィールド
請求元住所 2	G11	{{ds.publisher.address2}}	同上
請求元電話番号	H12	{{ds.publisher.tel}}	同上
請求元銀行名	G13	{{ds.publisher.bankname}}	同上
請求元銀行支店名	H13	{{ds.publisher.bankblanch}}	同上
請求元銀行口座番号	H14	{{ds.publisher.account}}	同上

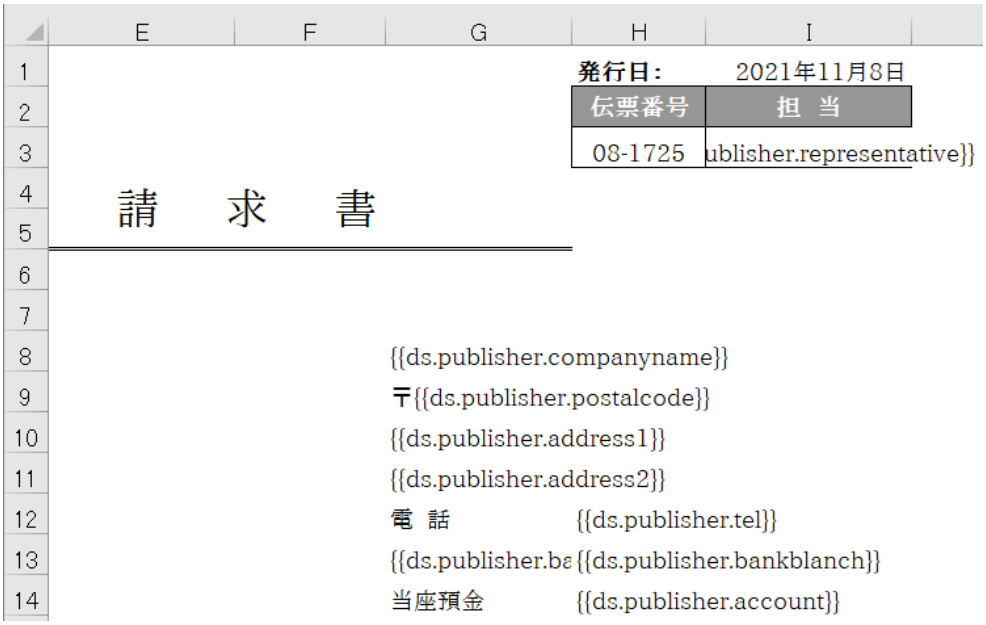
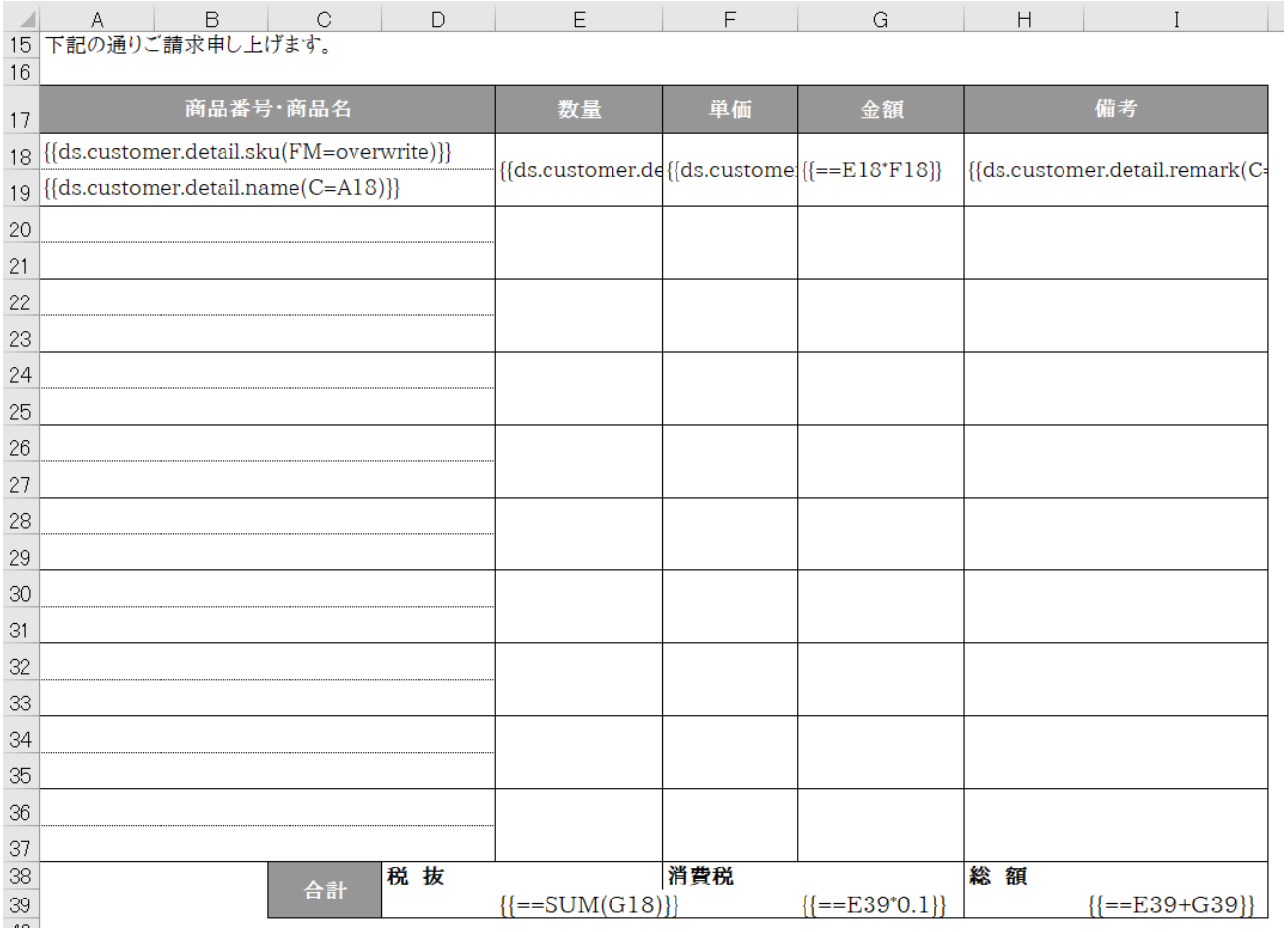


図 19　請求元情報

請求明細情報

請求明細（複数行）を表示する部分を作成します。データソースの **Detail** クラスで定義されるフィールドをテンプレート構文で記載します。

表示項目	セル位置	テンプレート構文	フィールド種別
商品番号	A18	{{ds.customer.detail.sku (FM=overwrite)}}	データフィールド（塗りつぶしモード=上書きを指定）
商品名	A19	{{ds.customer.detail.name (C=A18)}}	データフィールド（商品番号と対表示するためにカスタムコンテキストを指定）
数量	E18	{{ds.customer.detail.unit}}	データフィールド
単価	F18	{{ds.customer.detail.price}}	同上
金額	G18	{{==E18*F18}}	数式フィールド（式フィールドでも可）
備考	H18	{{ds.customer.detail.remark (C=A18)}}	データフィールド
合計（税抜）	E39	{{==SUM(G18)}}	数式フィールド（関数フィールドでも可）
合計（消費税）	G39	{{==E39*0.1}}	数式フィールド（式フィールドでも可）
合計（総額）	I39	{{==E39+G39}}	同上



	A	B	C	D	E	F	G	H	I
15	下記の通りご請求申し上げます。								
16									
17	商品番号･商品名				数量	単価	金額	備考	
18	{{ds.customer.detail.sku(FM=overwrite)}}				{{ds.customer.de	{{ds.custome	{{==E18*F18}}	{{ds.customer.detail.remark(C-	
19	{{ds.customer.detail.name(C=A18)}}								
20									
21									
22									
23									
24									
25									
26									
27									
28									
29									
30									
31									
32									
33									
34									
35									
36									
37									
38			合計	税　抜		消費税		総　額	
39				{{==SUM(G18)}}		{{==E39*0.1}}		{{==E39+G39}}	

図 20　請求明細情報

6. テンプレート処理の実行、Excel／PDF ファイル出力

Program.cs に「4. データソースの追加」で追加した **AddDataSource** メソッド以降の処理を追加します。

ファイル：Program.cs

```
// テンプレート処理を呼び出し
workbook.ProcessTemplate();

// Excel ファイルに保存
workbook.Save("result.xlsx");

// PDF ファイルに保存
workbook.Save("result.pdf");
```

アプリケーションの実行

ここまでの手順を終えるとプロジェクトは図 21 の画面のようになっています。ターミナルから **dotnet run** コマンドを入力してアプリケーションを実行します。

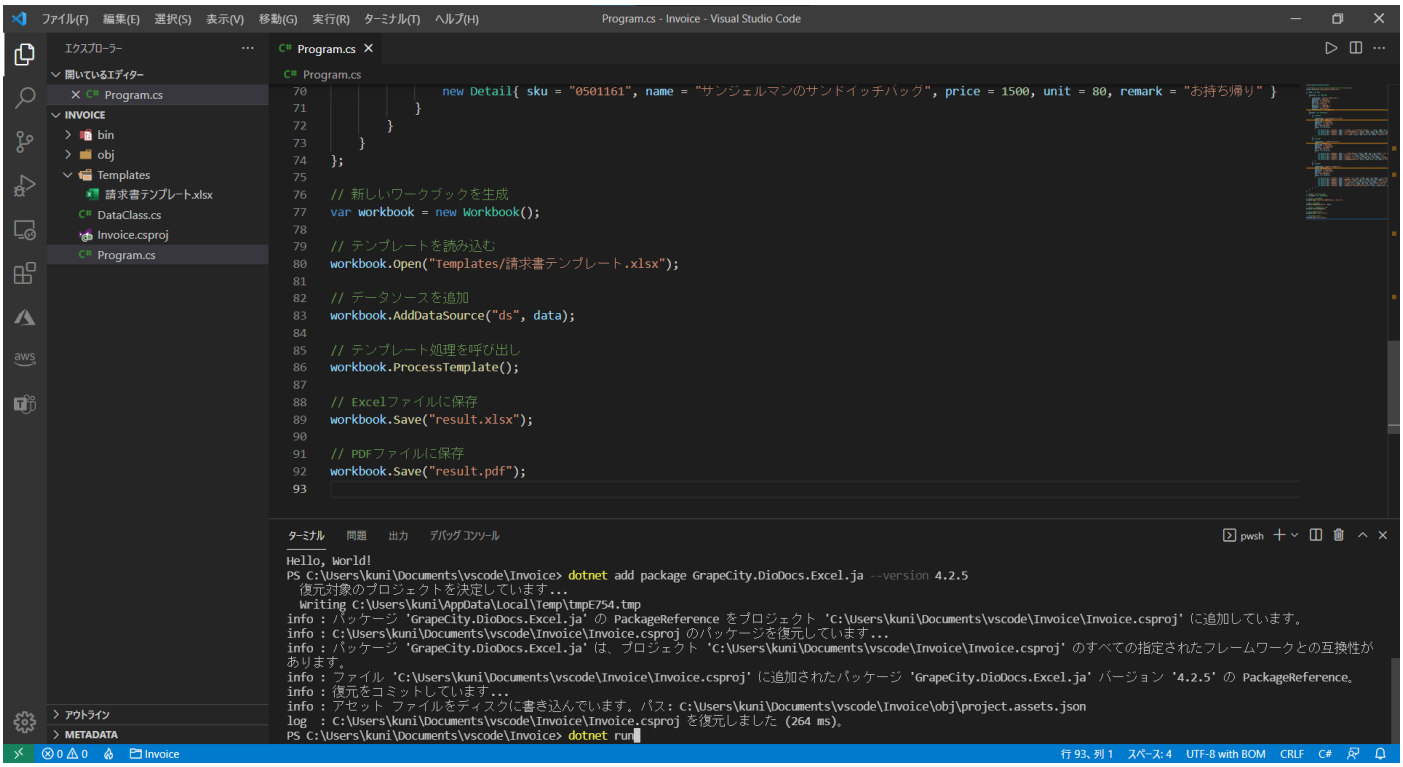


図 21　完成したプロジェクト（コンソールアプリケーション）

実行が完了すると「出力結果」で示した **result.xlsx** と **result.pdf** の 2 つのファイルが生成されます。

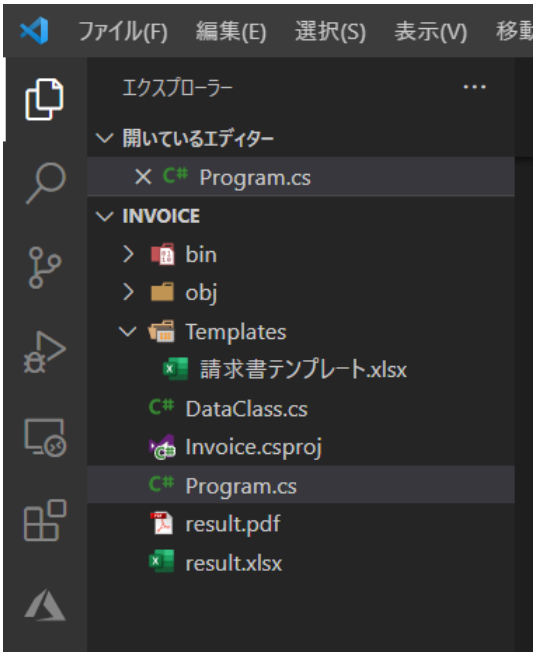


図 22　アプリケーションを実行後（result.xlxs と result.pdf が作成される）

請求書のカスタマイズ

前のセクション「請求書の作成」で作成した請求書（出力結果）を別のレイアウトで出力する例を幾つか示します。アプリケーション自体には手を加えません。変更するのは帳票テンプレートの Excel ファイルのみです。

ケース 1：可変明細の出力

データ数に関わらず 10 行固定の明細ではなく、データ数と同じ数の明細を出力する方法を紹介します。

帳票テンプレートの変更

請求明細情報を次のように変更します。

(1) 不要な空白行（20～37 行目）を削除する
(2) 商品番号（A18）から FM=overwirte を削除する

変更した帳票テンプレートは下記のとおりです。

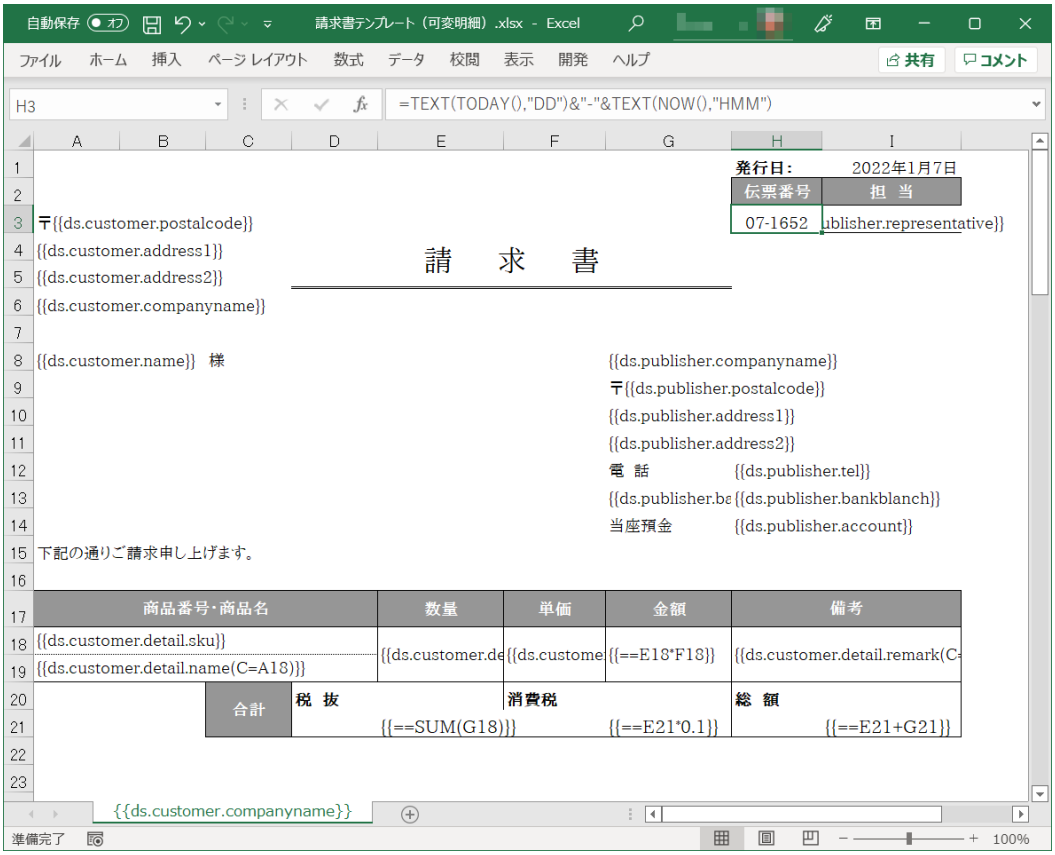


図 23　可変明細に対応させた帳票テンプレート

出力結果（Excel ファイル）

請求先によって明細数が異なっている場合にも対応できるようになります。

発行日: 2022年1月7日

伝票番号	担 当
07-1659	葡萄 城太郎

〒931-9999
M県S市広瀬区
花京院3-1-4
財団法人スピードワゴン

請 求 書

虎猿出井 富似雄 様

ディオドック株式会社
〒931-2222
M県S市紅葉区
杜王町2-6-11
電 話 022-777-3210
むかでや銀行 杜王町支店
当座預金 123-456789

下記の通りご請求申し上げます。

商品番号・商品名	数量	単価	金額	備考
0105231 モッツァレッラチーズとトマトのサラダ	70	¥1,200	¥84,000	前菜
0201071 塩鯖風スパゲティー	100	¥2,500	¥250,000	パスタ
0301201 小羊背肉のリンゴソースかけ	130	¥5,000	¥650,000	メイン
0401011 ごま蜜団子	60	¥800	¥48,000	デザート
合計	税 抜 ¥1,032,000	消費税 ¥103,200	総 額 ¥1,135,200	

図 24　可変明細の出力結果（1）

発行日: 2022年1月7日

伝票番号	担 当
07-1659	葡萄 城太郎

〒932-1234
M県S市秋保区
上愛子6-4-7
杜王グランドホテル

請 求 書

江陽 太郎 様

ディオドック株式会社
〒931-2222
M県S市紅葉区
杜王町2-6-11
電 話 022-777-3210
むかでや銀行 杜王町支店
当座預金 123-456789

下記の通りご請求申し上げます。

商品番号・商品名	数量	単価	金額	備考
0202061 激辛麻婆焼きそば	50	¥1,000	¥50,000	麺
0302031 塩竈マグロ	80	¥2,300	¥184,000	魚
0402301 超極厚牛タン	110	¥4,800	¥528,000	肉
0502021 濃厚ずんだロール	60	¥800	¥48,000	デザート
0602271 ハチミツパンケーキ	60	¥1,300	¥78,000	お持ち帰り
合計	税 抜 ¥888,000	消費税 ¥88,800	総 額 ¥976,800	

図 25　可変明細の出力結果（2）

ケース 2：請求先を単一シートにまとめる

請求先毎に別個のワークシートとするのではなく、請求先すべてを単一のワークシートに出力し、請求先毎にページを区切る方法を紹介します。

帳票テンプレートの変更

請求先情報を次のように変更します。

(1) シートタブからシート名フィールドを削除し、代わりに「請求書」という名称にします
(2) 請求先企業名（A6）にセル範囲として(R=A1:I40)を追加します
(3) A40 セルにページ区切りおよび A6 セルへのコンテキストを保持するフィールド{{(C=A6, Pb=true)}}を追加します
(4) 全ての請求先の請求明細が同一の高さとなるように、グローバル設定 **KeepLineSize** を true に設定します

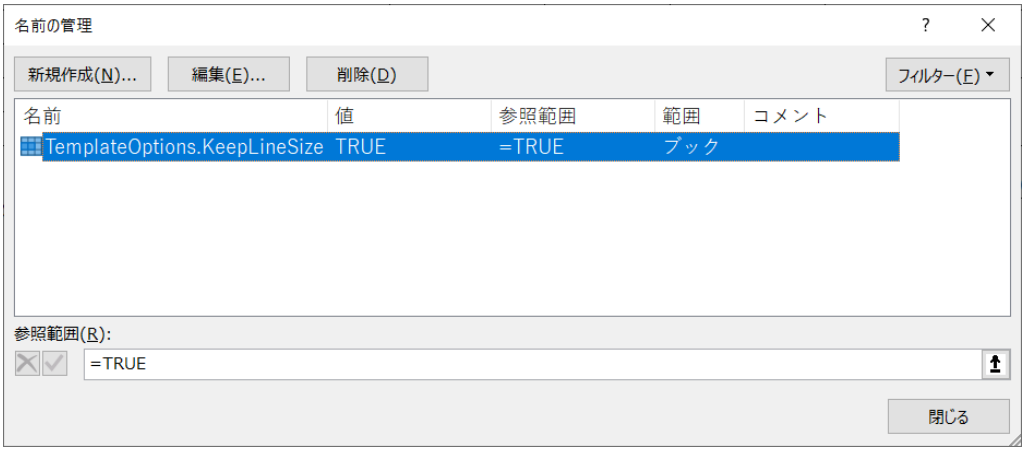


図 26　KeepLineSize の設定

KeepLineSize の詳細は製品ヘルプの以下のトピックを参照してください。
https://docs.grapecity.com/help/diodocs/excel/#global-settings.html

変更後の帳票テンプレートは下記のとおりです。

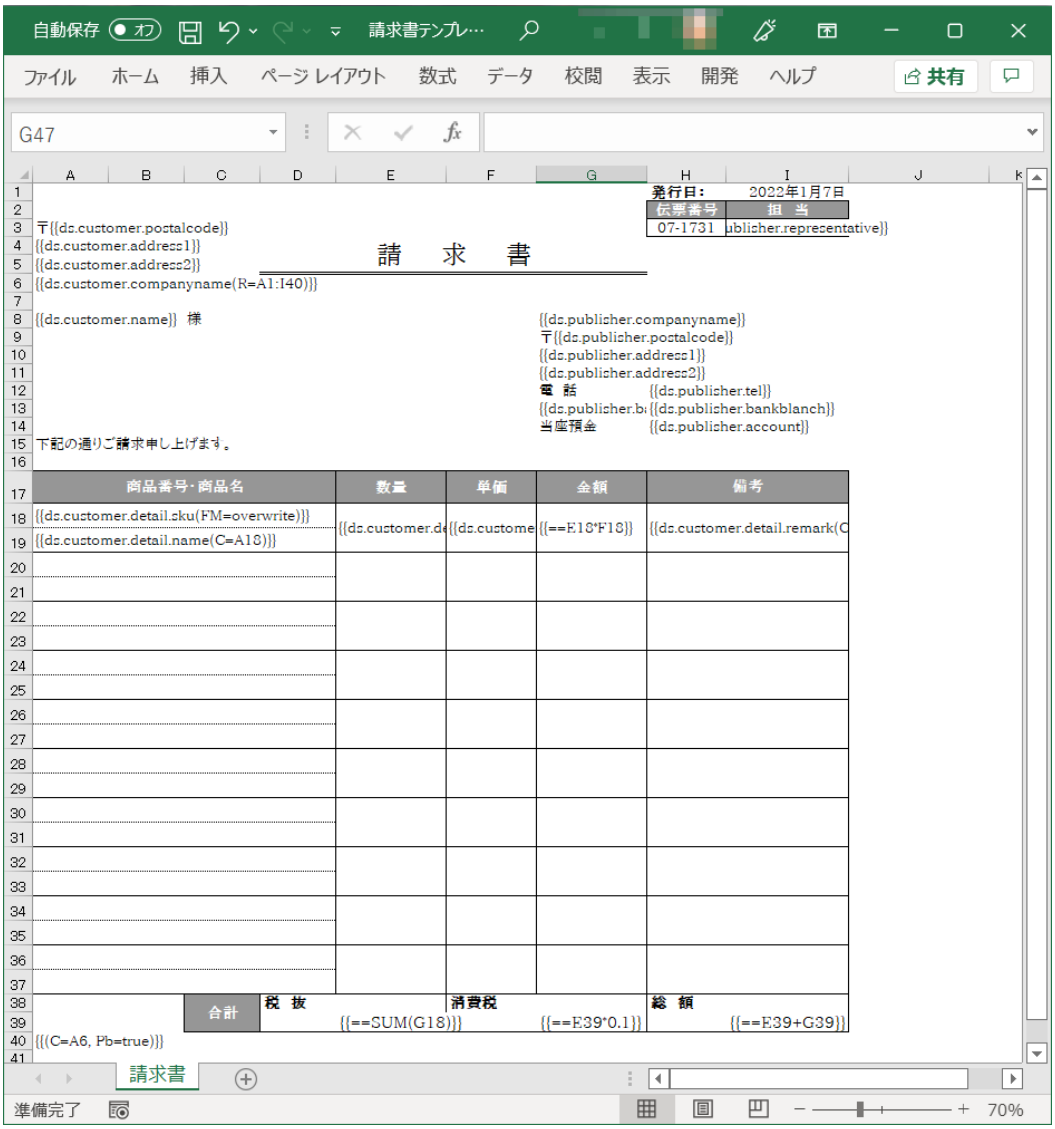


図 27　請求先を単一シートにまとめたテンプレート

出力結果（Excel ファイル）

単一のワークシートに全ての請求先が表示されます。また、請求先毎にページ区切りが挿入されます。

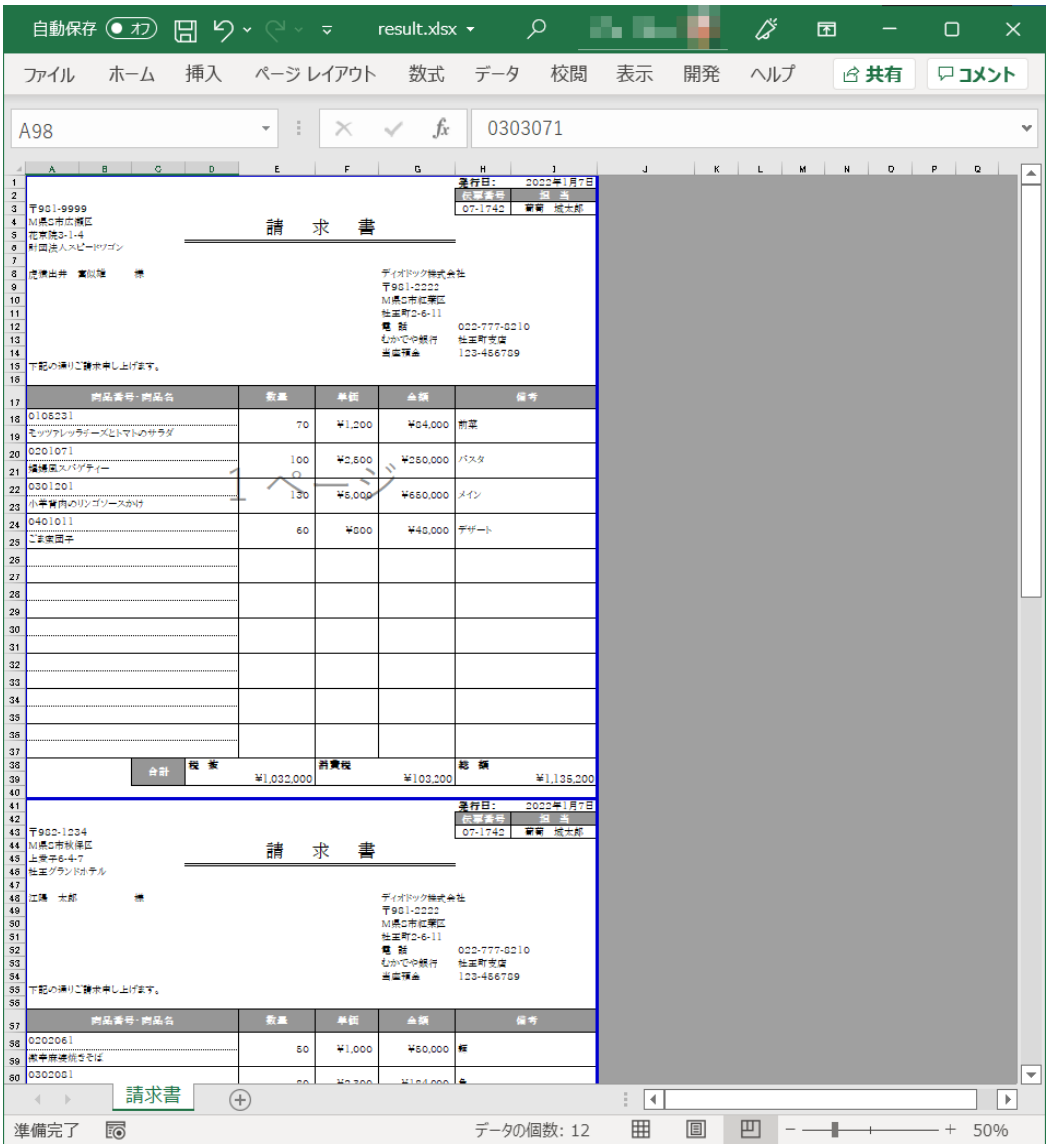


図 28　単一シートの出力結果（Excel ファイル）

出力結果（PDF ファイル）

請求先毎に改ページされたものが出力されます。

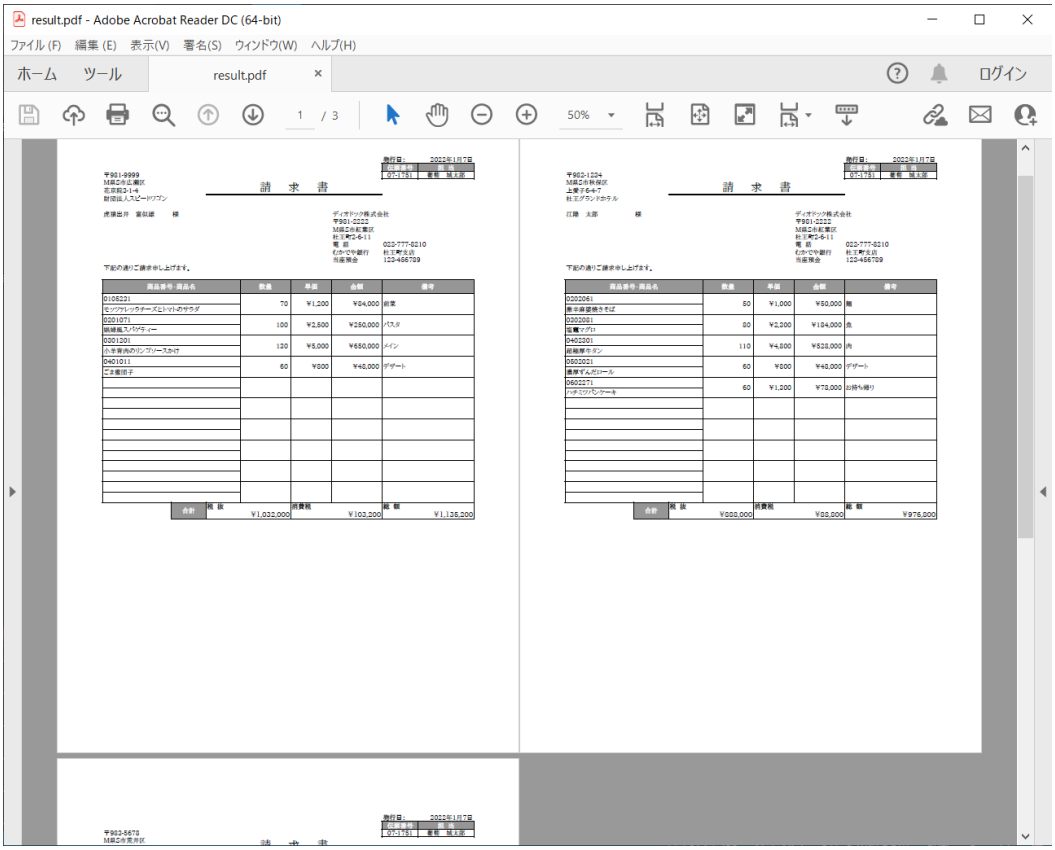


図 29　単一シートの出力結果（PDF ファイル）

ケース 3：備考欄とバーコードの追加

このケースではこれまでに触れていなかった少し高度な機能を使った例を 2 つ紹介します。どちらもテンプレート構文を使用したものですが、Excel 上では有効とならず PDF へ出力した場合にのみ有効となる機能です。

最初のカスタマイズは請求書の最下部に備考欄を作成します。この備考欄は Excel 上では単なるセルでしかありませんが、PDF に出力すると入力フォームに変化します。本機能の詳細は製品ヘルプの以下のトピックを参照してください。

https://docs.grapecity.com/help/diodocs/excel/#pdf-form-builder.html

2 番目のカスタマイズは商品番号データからバーコードを出力します。バーコードは Excel 上では単なる文字列ですが、PDF に出力するとバーコード画像に変化します。本機能の詳細は製品ヘルプの以下のトピックを参照してください。

https://docs.grapecity.com/help/diodocs/excel/#barcodes.html

帳票テンプレートの変更

(1) 42～47 行目に備考欄を追加し、その中にテキストボックスフィールドを表すテンプレート構文を記載します

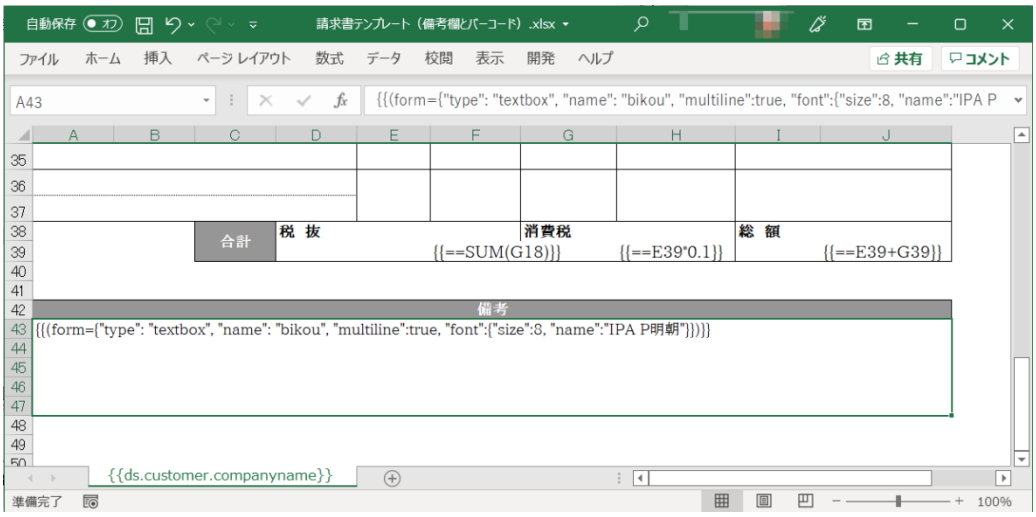


図 30　備考欄のテンプレート構文

(2) グローバル設定 **EmbedFontForFormFields** を false に設定します

注）デフォルトの true の場合、当該フォントの全グリフを埋め込むため PDF ファイルのサイズが大きくなります。false の場合、埋め込まれませんが、当該フォントが存在しない環境で PDF を表示・入力する際に期待した結果にならない場合があります。

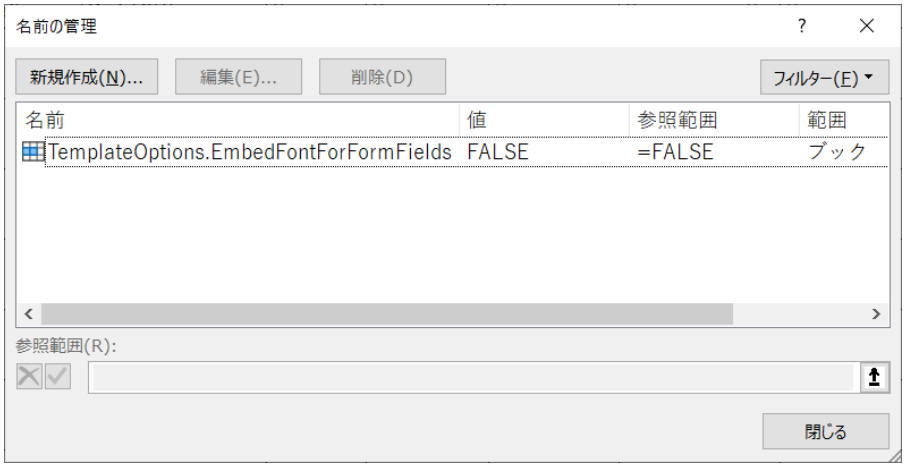


図 31　EmbedFontForFormFields の設定

EmbedFontForFormFields の詳細は製品ヘルプの以下のトピックを参照してください。
https://docs.grapecity.com/help/diodocs/excel/#global-settings.html

(3) レイアウトを変更し E18 セルにバーコード（EAN8）出力用のテンプレート構文を記載します

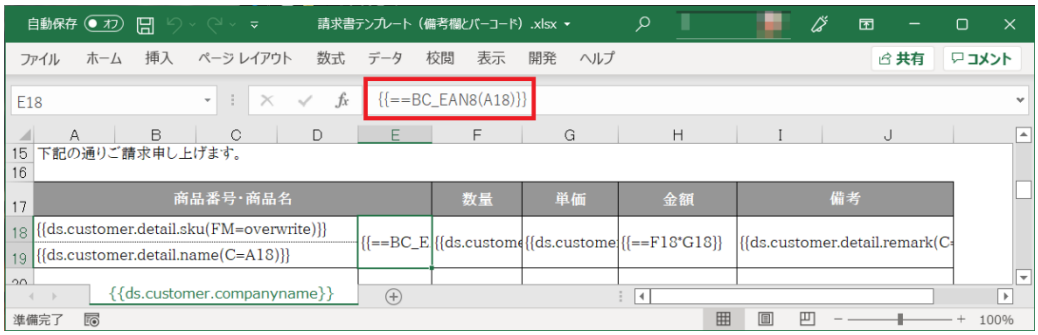


図 32　バーコードのテンプレート構文

出力結果（Excel ファイル）

備考欄のテンプレート構文は Excel 上では無視されます。バーコードを表すテンプレート構文は Excel 上では解釈できないので「**#NAME?**」というエラーが表示されます。

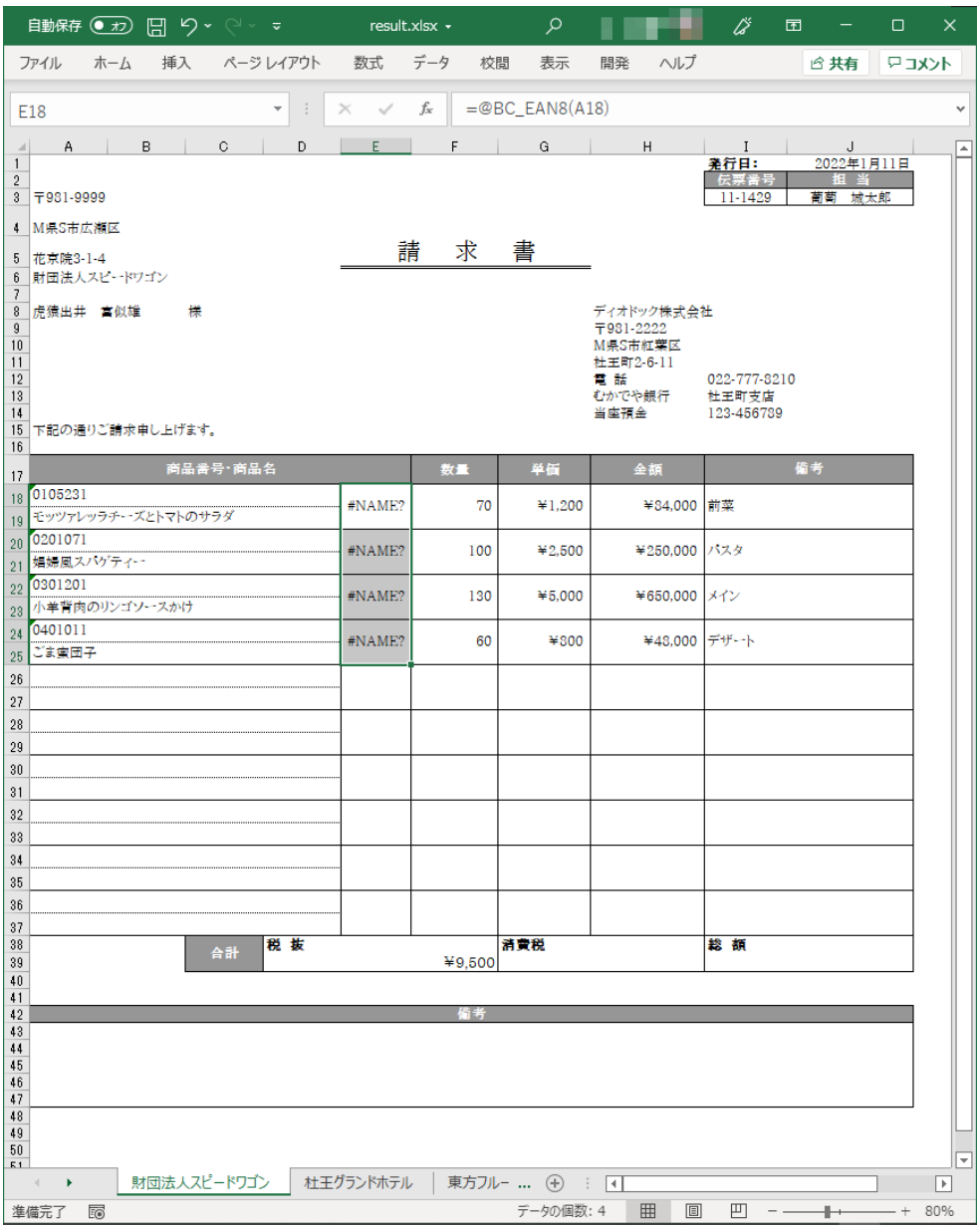


図 33　備考欄とバーコードの出力結果（Excel ファイル）

出力結果（PDF ファイル）

バーコードが表示されました。備考欄は PDF フォームとして作成されるため Adobe Reader などのビューワから直接文字を入力し上書き保存することができます。

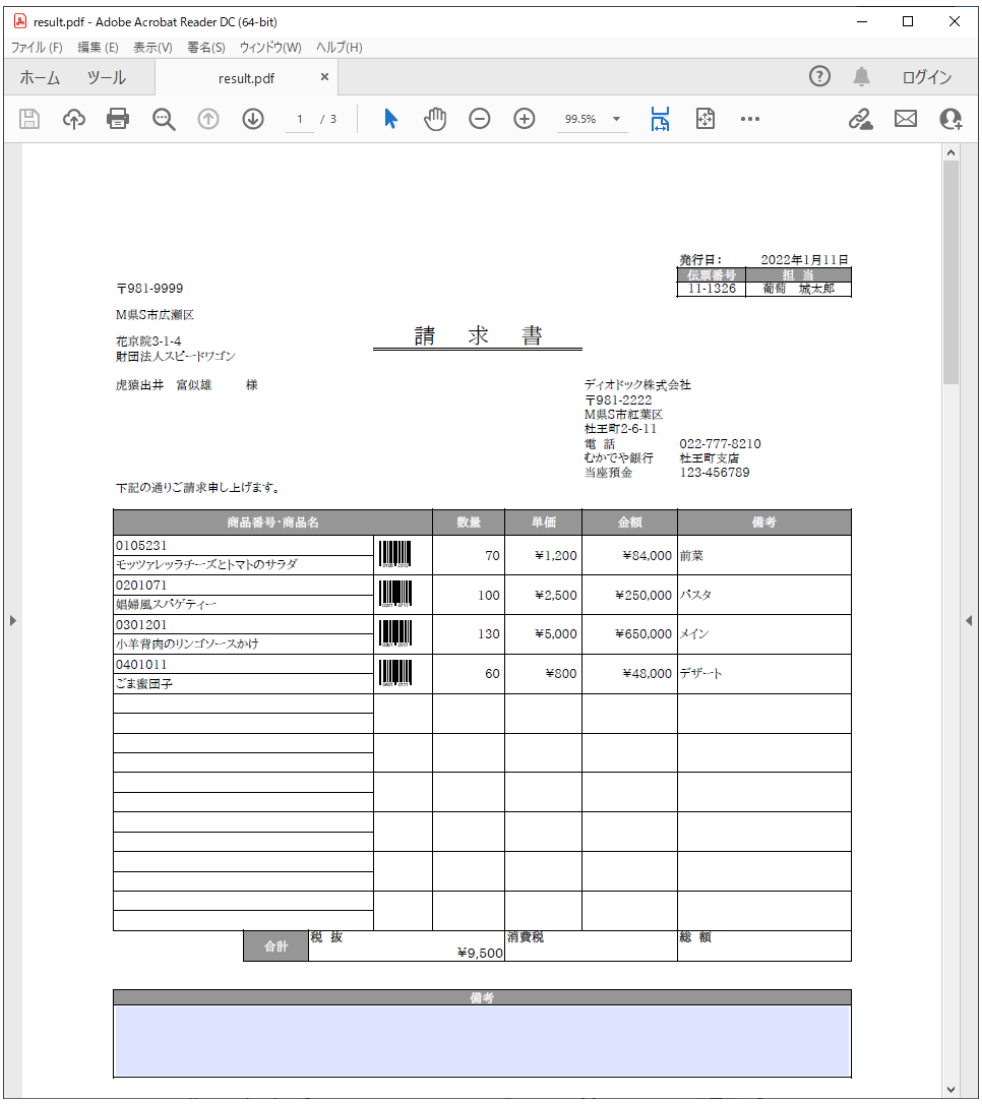


図 34　備考欄とバーコードの出力結果（PDF ファイル）