

# ASTERIA Warp

## デベロッパー・スキルチェックテスト

### 【注意事項】

- 制限時間は90分間です。
- 選択問題及び実習問題は指示されたWebフォームよりご回答ください。
- 合図があるまでは次頁を開かないようお願いします。
- 試験中参考書やヘルプなどの参照は問題ございません。
- Excelで表計算することは禁止行為となります。
- 試験中ご質問等ございましたら挙手されてスタッフをお呼びください。
- 試験終了の合図がありましたら速やかに解答をやめ、スタッフの指示に従ってください。
- デザイナーからサーバーへ接続する場合は、サーバー名「localhost」、ポート「21381」、ユーザー名「guest」、パスワード「guest」ドメイン「/」でログインしてください。

実習問題

1. 注文データ処理 全4問


【入力データ仕様】を参考に、ホームフォルダ配下にある「1/注文一覧(7月).csv」「1/顧客マスター.csv」「1/商品マスター.csv」について、以下の問いにASTERIA Warpを用いて回答してください。

- (1)商品コード「RTY070」の注文合計個数を回答してください。
- (2) 2019年7月21日の注文合計額を回答してください。
- (3) 30～39歳の顧客の注文について、分類番号ごとに注文個数を合計し、分類ランキングを作成してください。また、注文個数が最も多い商品の分類を回答してください。
- (4) 30～39歳の顧客の注文について、商品注文個数ランキングを作成します。  
(3)で求めた分類ランキングを元に、3番目に注文個数の多い分類の商品注文個数ランキングを以下の規則に従って作成してください。  
作成したCSVファイルをアップロードし、結果のレスポンスを回答してください。  
アップロードページ:<http://localhost:21380/aug/upload.html>

- 【規則】
- ・出力フォーマットは【出力データ 仕様】に従う
  - ・左から「商品コード」、「注文個数」の2フィールド
  - ・「商品コード」は、注文個数順(降順)に並べ、注文個数が同等の場合は商品コードのアルファベット順に並べる
  - ・「注文個数」は商品の合計注文個数とする
  - ・「No」に連番を付与

【例】

商品コード	注文個数
AAA111	2
BBB222	10
FFF666	2
DDD444	2
EEE555	7
CCC333	4
AAA111	1



No	商品コード	注文個数
1	BBB222	10
2	EEE555	7
3	CCC333	4
4	AAA111	3
5	DDD444	2
6	FFF666	2

## 【入力データ 仕様】

ホームフォルダ配下にある「1/注文一覧(7月).csv」「1/顧客マスター.csv」「1/商品マスター.csv」の仕様は以下の通りです。

なお、レコード開始行はすべて1行目です。

- 注文一覧(7月).csv

項目	データ型	備考
注文ID	Integer	5桁の数字
日付	String	yyyy/MM/dd 2019年7月のデータのみ
商品コード	String	
顧客ID	Integer	8桁の数字
個数	Integer	

- 顧客マスター.csv

項目	データ型	備考
顧客ID	Integer	8桁の数字
氏名	String	姓名の間は半角スペース
氏名(カタカナ)	String	姓名の間は半角スペース
性別	String	
メールアドレス	String	
年齢	Integer	

- 商品マスター.csv

項目	データ型	備考
商品コード	String	
分類番号	Integer	1～9の数字
商品名	String	
価格	Integer	

## 【出力データ 仕様】

- 注文個数ランキング.csv

項目	データ型	備考
No	Integer	連番
商品コード	String	
注文個数	Integer	

実習問題

2. 株価データ取得 全4問

株価を自動取得するシステムを作成します。以下の問いにASTERIA Warpを用いて回答してください。※ 株価は架空のデータです。  
API仕様、CSVファイル仕様はそれぞれ【サイトA API仕様】、【CSVファイル仕様】を参照してください。

- (1) 株価を掲載しているサイトAからWebスクレイピングを行って、ある会社の株価データを自動取得します。取得できる株価のうち、高値と安値の差が最も大きい日付をyyyy/MM/ddの形式で回答してください。  
サイトAのURL: <http://localhost:21380/aug/stock/3886>

株価 [3886]				
日付	始値	高値	安値	終値
2019年7月31日	1,026	1,048	1,021	1,045
2019年7月30日	1,012	1,025	998	1,015
2019年7月29日	1,037	1,043	996	996
2019年7月26日	1,084	1,085	1,036	1,048
2019年7月25日	1,058	1,072	1,043	1,070
2019年7月24日	1,049	1,060	1,034	1,050
2019年7月23日	1,074	1,074	1,051	1,051
2019年7月22日	1,086	1,099	1,058	1,073
2019年7月19日	1,127	1,127	1,070	1,091
2019年7月18日	1,100	1,157	1,100	1,126
2019年7月17日	1,112	1,116	1,081	1,097
2019年7月16日	1,137	1,144	1,115	1,118
2019年7月12日	1,118	1,146	1,108	1,145
2019年7月11日	1,097	1,114	1,084	1,103
2019年7月10日	1,100	1,107	1,077	1,101
2019年7月9日	1,044	1,109	1,037	1,096
2019年7月8日	1,050	1,074	1,044	1,051

【ヒント】  
HtmlParseコンポーネントを利用してHTMLの解析を行うことができます。  
また、ある要素Eのn番目を取得するには以下のように指定します。  
E:nth-of-type(n)

- (2) サイトAにおいて、REST APIの提供が開始されたため、APIを利用して株価を自動取得します。銘柄コード2895の2019年7月17日の高値を回答してください。
- (3) APIを利用して銘柄コード2895の2019年7月1日～31日までの株価のうち、始値が前日終値と比べ最も大きく増加している日付をyyyy/MM/ddの形式で回答してください。なお、2019年7月1日の前日株価終値は "1010"とし、7月の取引日はguestユーザーのホームフォルダ配下にある「2/dates.csv」のデータを参照してください。
- (4) サイトAには銘柄コードを任意のグループ別にお気に入り登録する機能があります。guestユーザーのホームフォルダ配下にある「2/favs.csv」のデータを元に、APIを利用して銘柄コードとグループを登録してください。また、登録完了後のレスポンスデータのうち、コード番号を回答してください。

【CSVファイル仕様】

• dates.csv

項目	データ型	備考
日付	String	yyyyMMdd

• favs.csv

項目	データ型
銘柄コード	Integer
グループ番号	String

## 【サイトA API仕様】

### 【共通仕様】

	項目	説明・値
概要	使用するプロトコル	HTTP
	文字コード	UTF-8
リクエストヘッダ	X-ApplicationKey	guestユーザーのホームフォルダ配下にある「2/key.txt」の値

### 【株価取得】

項目	説明・値												
リクエストURL	http://localhost:21380/aug/getstock/{銘柄コード}												
HTTPメソッド	GET												
パラメーター	<table><tr><td>date</td><td>取得する株価の日付 yyyyMMdd形式の文字列</td></tr></table>		date	取得する株価の日付 yyyyMMdd形式の文字列									
date	取得する株価の日付 yyyyMMdd形式の文字列												
レスポンス	<table><tr><td rowspan="5">{   "record":{     "date":“日付”,     "open":“始値”,     "high":“高値”,     "low":“安値”,     "close":“終値”   } }</td><td>record.date</td><td>日付</td></tr><tr><td>record.open</td><td>始値</td></tr><tr><td>record.high</td><td>高値</td></tr><tr><td>record.low</td><td>安値</td></tr><tr><td>record.close</td><td>終値</td></tr></table>		{ "record":{ "date":“日付”, "open":“始値”, "high":“高値”, "low":“安値”, "close":“終値” } }	record.date	日付	record.open	始値	record.high	高値	record.low	安値	record.close	終値
{ "record":{ "date":“日付”, "open":“始値”, "high":“高値”, "low":“安値”, "close":“終値” } }	record.date	日付											
	record.open	始値											
	record.high	高値											
	record.low	安値											
	record.close	終値											

### 【お気に入り登録】

項目	説明・値								
リクエストURL	http://localhost:21380/aug/favstocks								
HTTPメソッド	POST								
ボディ	<div><pre>{   "version": "1.0",   "favs": [     { "code": "銘柄コード", "group": "グループ番号" },     { "code": "銘柄コード", "group": "グループ番号" }   ] }</pre></div> <p>※すべて必須項目</p> <table><tr><td>version</td><td>APIバージョン "1.0"を指定</td></tr><tr><td>favs</td><td>お気に入り情報</td></tr><tr><td>favs.code</td><td>銘柄コード</td></tr><tr><td>favs.group</td><td>グループ番号</td></tr></table>	version	APIバージョン "1.0"を指定	favs	お気に入り情報	favs.code	銘柄コード	favs.group	グループ番号
version	APIバージョン "1.0"を指定								
favs	お気に入り情報								
favs.code	銘柄コード								
favs.group	グループ番号								
レスポンス	<pre>{   "code": "コード",   "message": "メッセージ" }</pre>								