

Úloha 3

List obsahuje **vzorové řešení úlohy 3**

Zadání

Vytvořte uuScript, který pro vygenerovaný pool zaměstnanců z úlohy 1 graficky prostřednictvím grafů zobrazí:

- rozložení pracovních úvazků formou koláčového grafu,
- četnost věku všech mužů formou sloupcového grafu,
- četnost pracovních úvazků mužů a žen formou složeného sloupcového grafu.

i Jako vstupní data použijte výstupní data, která jste vygenerovali v úloze 1.

i Zvažte vhodné použití vlastních metod, tzv. helperů. Např. vytvoření helperu pro určení věku na základě data narození.

i Grafy zobrazte s využitím knihovny uu5 Chart.



uu5Chart g01 - User Guide



Ukázka struktury vstupních dat (tzv. dtoIn):

```
const dtoIn = [  
  {  
    gender: "male",  
    birthdate: "2000-08-07T00:00:00.000Z",  
    name: "Jan",  
    surname: "Novák",  
    workload: 40  
  },  
  {  
    gender: "female",  
    birthdate: "2000-01-03T00:00:00.000Z",  
    name: "Jana",  
    surname: "Nováková",  
    workload: 20  
  }  
];
```

Validační schéma vstupních dat:

```
const sdeExamTask03DtoInSchemaType = array(  
  shape({  
    gender: oneOf("male", "female").isRequired(),  
    birthdate: date().isRequired(),  
    name: string().isRequired(),  
    surname: string().isRequired(),  
    workload: oneOf(40, 30, 20, 10).isRequired()  
  }), 2000)
```



Dokumentaci či učebnice, kterou můžete v průběhu řešení úloh využítí, naleznete [zde](#).

Design

Error Prefix: ucl/sdeExamTask03/

1. **Step** // Inicializace

Inicializace objektu `uuAppErrorMap`, který je vytvořen jako prázdný objekt a je připraven pro ukládání chyb a upozornění.

```
uuAppErrorMap = {}
```

2. Sequence // Validace dtoIn

Validace vstupních dat prostřednictvím standardní metody `Validator.validate`, které jsou předána vstupní data (`dtoIn`) a validační schéma `dtoInType`. Validátor vyplní `validationResult`, který je propsán do `uuAppErrorMap`.

Typicky nastávají tyto čtyři případy:

- datový typ klíče - chyba (error)
- hodnota klíče - chyba (error)
- zda-li je klíč povinný - chyba (error)
- zda-li je klíč očekávaný - neočekávané klíče jsou smazány - upozornění (warning).

Validace je řešena helperem `_validateDtoIn(dtoIn, uuAppErrorMap)`.

2.1. Sequence

Standardní validace `dtoIn`.

2.1.1. Step

Zavolání metody `validate` pro `dtoIn` s využitím `dtoInType` a následně vyplnění `validationResult` výsledkem validace.

2.1.2. Sequence

Zavolání `processValidationResult` a zapsání `validationResults` do `uuAppErrorMap`.

2.1.2.1. Warning

Zapsání upozornění do `uuAppErrorMap`, pokud existují ve vstupních datech nadbytečné klíče (tzv. `unsupported keys`).

`unsupportedKeys`

`DtoIn contains unsupported keys.`

```
1  unsupportedKeyList:["..."] //unsupported dtoIn keys were deleted
```

2.1.2.2. Error

Zapsání chyby do `uuAppErrorMap` a ukončení vykonání skriptu, pokud nejsou vstupní data validní.

`invalidDtoIn`

`DtoIn is not valid.`

```
1  invalidTypeKeyMap:["..."] //map of keys with invalid type
2  invalidValueKeyMap:["..."] //map of keys with invalid value
3  missingKeyMap:["..."] //map of required keys
```

2.1.3. Step

Vrácení `uuAppErrorMap`.

Implementace



Unit test úlohy 3 spustíte vnořené složce `uun_sde_hackaton_employees` příkazem `npm test task3.test.js`.



Pro zobrazení koláčového grafu můžete použít připravený helper `_getPieChart(headerText, valueLabel, data)`. Použití je znázorněno v následujícím kódu.

```
1  const dataPieChart = [  
2    {  
3      label: "10h/týden",  
4      value: workload10  
5    },  
6    {  
7      label: "20h/týden",  
8      value: workload20  
9    },  
10   {  
11     label: "30h/týden",  
12     value: workload30  
13   },  
14   {  
15     label: "30h/týden",  
16     value: workload40  
17   }  
18 ];  
19  
20 const pieChart = _getPieChart("Rozložení pracovních úvazků", "Zaměstnanci",  
    dataPieChart);
```



Pro zobrazení sloupcového grafu můžete použít připravený helper `_getBarChart(headerText, valueLabel, data)`. Použití je znázorněno v následujícím kódu.

```
1  //ukázková struktura dat  
2  const dataBarChart = [  
3    {"label": "19", "value": 3},  
4    {"label": "23", "value": 5}  
5  ];  
6  
7  const barChart = _getBarChart("Histogram věku mužů", "Zaměstnanci", dataBarChart
```

V případě skládaného sloupcového grafu pak můžete použít helper `_getBarChartStacked(headerText, data)`. Použití je znázorněno v následujícím kódu.

```
1  //ukázková struktura dat  
2  const dataBarChartStacked = [  
3    {"label": "19", "valueMale": 2, "valueFemale": 1},  
4    {"label": "23", "valueMale": 3, "valueFemale": 2},  
5  ];  
6  
7  const barChartStacked = _getBarChartStacked("Histogram pracovních úvazků zaměstn  
    dle pohlaví", dataBarChartStacked);
```

