Домашнее задание к лекции 2.4 «Прототип и конструктор функции»

Перед началом работы

1.Активируйте строгий режим соответствия.

2.Скопируйте код к себе в редактор:

```
const items = [
  {
    title: 'Телепорт бытовой VZHIH-101',
    available: 7,
    holded: 0
  },
  {
    title: 'Ховерборд Mattel 2016',
    available: 4,
    holded: 5
  },
  {
    title: 'Меч световой FORCE (синий луч)',
    available: 1,
    holded: 1
  }
];
```

Задача № 1. Продажа со склада и из резерва «в долг».

Перед началом работы добавьте этот код в редактор:

```
const itemPrototype = {
  sell(field, amount = 1) {
    if (this[field] < amount) {
      throw `Недостаточно товара для продажи (${this[field]} из ${amount})`
    }
    this[field] -= amount;
    return true;
  },
```

```
  sellHolded(amount = 1) {
    return itemPrototype.sell.call(this, 'holded', amount);
  },
  sellAvailable(amount = 1) {
    return itemPrototype.sell.call(this, 'available', amount);
  }
};
```

Нам нужно обновлять остатки товаров на складе при продаже товара. У нас уже есть функционал для этого в объекте `itemPrototype`. Но, как говорит наш ведущий разработчик, мы не можем вносить изменения в объекты из массива `items`, поэтому требуется найти другое решение, которое не затронет товары, и при этом задействует уже существующий функционал.

Описание функции

Функция `sellItem` должна принимать товар из массива `items` или аналогичный и обновлять его остатки и резерв, используя функции `itemPrototype.sellAvailable`, если продажа осуществляется из остатка или `itemPrototype.sellHolded`, если продажа осуществляется из резерва. Функция сама не должна никак изменять объект товара. Принимает следующие аргументы:

1. `item` — товар, объект;
2. `amount` — количество товара, которое требуется зарезервировать, целое число;
3. `isHolded` — нужно ли списывать из резерва, по умолчанию `false`, логический тип.

Пример использования функции

```
sellItem(items[2], 1);
console.log(items[2].available); // 0
console.log(items[2].holded); // 1

sellItem(items[1], 4, true);
console.log(items[1].available); // 4
console.log(items[1].holded); // 1

const item = { available: 0, holded: 1 };
sellItem(item, 1, true);
console.log(item.available); // 0
console.log(item.holded); // 0
```

Убедитесь, что все примеры в вашей реализации дают именно такой результат. И попробуйте свои варианты использования.

Процесс реализации

1. Создайте функцию `sellItem`.

2.Проверьте аргумент `isHolded`: если он равен `true`, воспользуйтесь функцией `itemPrototype.sellHolded`, иначе — `itemPrototype.sellAvailable`.
3.Вызовите выбранную функцию в контексте объекта, переданного в `item`, передав в неё количество из аргумента `amount`.

Задача № 2. Форматированный вывод списка.

Перед началом работы добавьте код в редактор:

```
function formatFull() {
  return `${this.title}:\n\tдоступно ${this.available} шт.\n\tв резерве $
{this.holded} шт.`;
}

function formatLite() {
  return `${this.title} (${this.available} + ${this.holded})`;
}

function show(format) {
  console.log(format());
}
```

В разных разделах системы нам нужно выводить список товаров в разном виде. Поэтому нужна функция, которая бы выводила каждый товар, используя функцию `show`, отформатировав товар заданной функцией, например, `formatFull` или `formatLite`.

Описание функции

Реализовать функцию `showItems`, которая будет принимать список товаров, аналогичный массиву `items`, и выводить каждый элемент списка, используя функцию show и переданную функцию форматирования товара. Принимает следующие аргументы:

1.`list` — список товаров, массив;
2.`formatter` — функция форматирования, функция. Функция не должна менять объекты в массиве `list` и сама что-либо выводить в консоль. Обратите внимание на то, что функция `show` принимает функцию, которая должна вернуть строку.

Пример использования функции

```
showItems(items, formatFull);
console.log('---');
showItems(items, formatLite);
```

Если функция showItems реализована верно, то вывод будет таким:

```
Телепорт бытовой VZHIH-101:
        доступно 7 шт.
```

```
        в резерве 0 шт.
Ховерборд Mattel 2016:
        доступно 4 шт.
        в резерве 1 шт.
Меч световой FORCE (синий луч):
        доступно 0 шт.
        в резерве 1 шт.
---
Телепорт бытовой VZHIH-101 (7 + 0)
Ховерборд Mattel 2016 (4 + 1)
Меч световой FORCE (синий луч) (0 + 1)
```

Процесс реализации

1. Создайте функцию `showItems`.
2. Пролистайте список из аргумента `list` оптимальным способом.
3. Для каждого элемента списка вызовите функцию `show`, передав в неё функцию, которая отформатирует товар, используя функцию из аргумента `formatter`.

Проверьте работу функции по примерам использования. А также попробуйте с её помощью вывести свой список товаров и свою функцию форматирования товара.

Задача № 3. Кнопка «Купить».

Перед началом работы скопируйте код в редактор:

```
function createButton(title, onclick) {
  return {
    title,
    onclick,
    click() {
      this.onclick.call(this);
    }
  };
}
```

Описание функции

Создайте функцию `createBuyButtons`, которая будет принимать список товаров, и для каждого товара из списка создаст кнопку с заголовком Купить, используя функцию `createButton`. При «клике» на кнопку для товара с названием `Телепорт бытовой VZHIH-101` в консоль должно выводиться `Телепорт бытовой VZHIH-101 добавлен в корзину`. Функция принимает следующий аргумент:

1. `items` — список товаров, аналогичный `items`, массив.

Функция должна вернуть массив кнопок, созданных функцией `createButton`.
Функция `createButton` принимает название кнопки и функцию, которая вызывается при «клике» на кнопку. Клик на кнопке симулируется вызовом метода `click` у созданной кнопки.

Пример использования функции

```
const buttons = createBuyButtons(items);
buttons[0].click();
buttons[2].click();
buttons[1].click();
```

Если функция `createBuyButtons` реализована верно, то вы получите такой вывод в консоль:

```
Телепорт бытовой VZHIH-101 добавлен в корзину
Меч световой FORCE (синий луч) добавлен в корзину
Ховерборд Mattel 2016 добавлен в корзину
```

Процесс реализации

1. Создайте функцию `createBuyButtons`.
2. Пролистайте список из аргумента `items` оптимальным способом.
3. Для каждого элемента списка вызовите функцию `createButton`, передав первым аргументом строку Купить, а вторым – функцию, которая выведет при вызове название товара в консоль.

Убедитесь, что пример использования функции работает как описано выше.