

1) Створити карту користувача(User Card):

Створити функцію «**userCard**» яка приймає число(будь-яке число) і повертає об'єкт з методами:

```
const card3 = userCard(3); // returns an object with methods
```

User Card методи:

getCardOptions. Дана функція повертає об'єкт котрий містить інформацію про карту:

```
card3.getCardOptions(); // returns options object with card info
```

Об'єкт має містити такі властивості:

- **balance** (по замовчуванню 100)
- **transactionLimit** (по замовчуванню 100. Це ліміт коштів які ви можете взяти з карти)
- **historyLogs** (масив об'єктів який містить інформацію про операції та транзакції даної карти)
- **key** (число в діапазоні[1; 3] залежить від числа яке ви передали в userCard функції. Кожна карта повинна мати унікальний **key**)

putCredits. Ця функція отримує певну кількість грошей і заповнює баланс карти:

```
card3.putCredits(150);
```

takeCredits. Ця функція отримує певну кількість грошей і віднімає ці кошти з балансу карти (протилежний за дією від метода **putCredits**):

```
card3.takeCredits(100);
```

Ви можете брати кошти з картки, якщо ліміт транзакцій та залишок коштів перевищують кількість коштів, які ви бажаєте взяти. Якщо ні, то слід записати відповідне повідомлення у консолі (скористайтесь console.error)

setTransactionLimit.Ця функція приймає як аргумент число і встановлює його як ліміт транзакцій на картці

```
card3.setTransactionLimit(5000);
```

transferCredits.

Ця функція отримує два аргументи - кількість кредитів, які потрібно передати, та карту одержувача (інший об'єкт, створений функцією `userCard` - card1):

```
card3.transferCredits(50, card1);
```

Кошти, які ви хочете передати, повинні обкладатися податками 0,5%.

Не забудьте перед перерахуванням перевірити залишок балансу та ліміт транзакцій відправника кредитів (card3).

Ви повинні відслідковувати **тільки Put credits/Take credits/Transaction limit change** операції і зберігайте history log. History log (Дивитись зображення 1) інформація має зберігатись в об'єктах з такими властивостями:

- **operationType** (стрічка яка описує здійснену операцію)
- **credits** (кількість коштів)
- **operationTime** (формат: "27/07/2018,03:24:53". Час коли була здійснена операція)

```
> card1.getCardOptions()
< {key: 2, balance: 150, transactionLimit: 200, historyLogs: Array(3)}
  balance: 150
  historyLogs: Array(3)
    0: {operationType: "Received credits", credits: 100, operationTime: "14/02/2019, 17:59:33"}
    1: {operationType: "Withdrawal of credits", credits: 50, operationTime: "14/02/2019, 17:59:44"}
    2: {operationType: "Transaction limit change", credits: 200, operationTime: "14/02/2019, 17:59:59"}
    length: 3
  __proto__: Array(0)
  key: 2
  transactionLimit: 200
  __proto__: Object
```

Зображення 1 — Приклад виклику функції

getCardOptions

2) Створити UserAccount:

Створити клас `UserAccount` (для цього завдання можна використати ES6 клас або звичайну функцію):

```
const user = new UserAccount('Bob');
```

Клас має містити:

Властивості акаунту користувача:

- **name** (передати в конструкторі)
- **cards** (Масив карток користувача. Масив не має бути довшим ніж 3)

Методи акаунту користувача:

- **addCard**. Створює нову карту (використовуйте 'userCard' функцію) та додайте її до cards:

```
user.addCard();
```

Користувач повинен мати ≤ 3 карти.

- **getCardByKey**. Приймає число в діапазоні {1; 3} і повертає об'єкт карти

```
/*
 * Повертає об'єкт {
 *   key: 1,
 *   balance: 150,
 *   ...other card properties
 * }
 */
user.getCardByKey(1);
```

Код

Приклад переказу коштів:

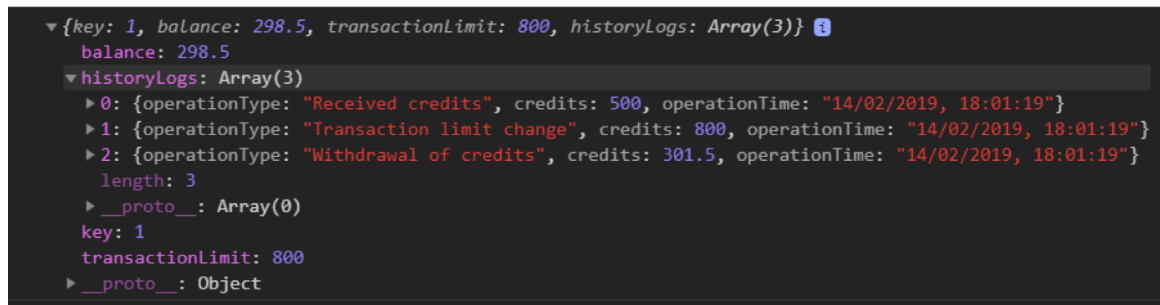
```
let user = new UserAccount('Bob');
user.addCard()
user.addCard()
```

```
let card1 = user.getCardByKey(1);
let card2 = user.getCardByKey(2);
```

```
card1.putCredits(500);
card1.setTransactionLimit(800);
card1.transferCredits(300, card2);
```

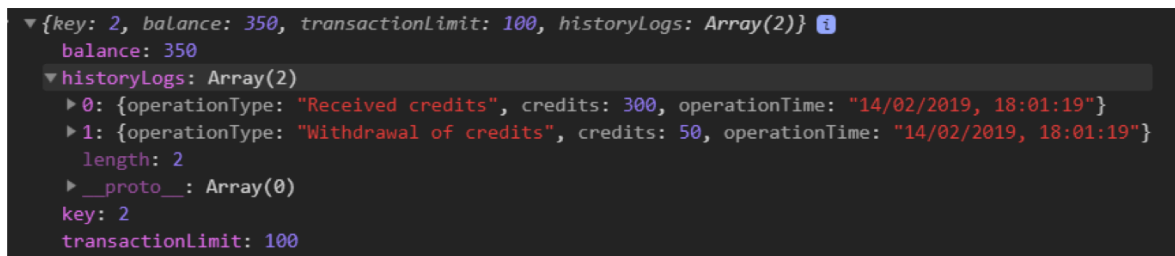
```
card2.takeCredits(50);
```

```
console.log(card1.getCardOptions()); // див зображення 2
```



Зображення 2 - Об'єкт властивостей card1

```
console.log(card2.getCardOptions()); // див зображення 3
```



Зображення 3 - Об'єкт властивостей card2

Посилання

- <https://css-tricks.com/javascript-scope-closures/>
- <https://developer.mozilla.org/uk/docs/Web/JavaScript/Reference/Classes>