

Міністерство освіти і науки України
Хмельницький національний університет



ЗБІРНИК НАУКОВИХ ПРАЦЬ
за матеріалами XIV Всеукраїнської науково-практичної конференції
«Актуальні проблеми комп'ютерних наук АПКН-2022»

18-19 листопада 2022

Хмельницький 2022

УДК 004:37:001:62

Збірник наукових праць за матеріалами XIV Всеукраїнської науково-практичної конференції «Актуальні проблеми комп'ютерних наук АПКН-2022». Хмельницький – 2022. – 331с.

У збірнику наукових праць подані перспективні практичні розробки аспірантів, студентів та здобувачів в області сучасних інформаційних технологій. Розглянуто актуальні проблеми комп'ютерних наук, комп'ютерної інженерії, прикладної математики й інженерії програмного забезпечення, приведено ряд робіт по впровадженню інформаційних технологій у виробництво та управління. Висвітлено перспективні розробки сучасних систем пошуку, обробки й захисту інформації, медійних та комунікаційних системи.

УДК 004:37:001:62

Матеріали конференції відтворені з авторських оригіналів. При макетуванні можливі незначні зміни компоновки контенту авторських оригіналів.

Участь у конференції та складові всіх її етапів (розгляд праць, макетування, публікація збірника наукових праць та видача сертифікатів) є безкоштовними для всіх учасників. Оргкомітет конференції висловлює подяку учасникам конференції та сподівається на подальшу співпрацю.

З питань проведення конференції та подальшого обміну інформацією звертатись на e-mail конференції: apkt.khnu@gmail.com

АКТУАЛЬНІ ПРОБЛЕМИ КОМП'ЮТЕРНИХ НАУК - 2022

XIV Всеукраїнська науково-практична конференція

Метою конференції є висвітлення актуальних проблем комп'ютерних наук, інформатики та інформаційних технологій.

СЕКЦІЇ КОНФЕРЕНЦІЇ:

1. Комп'ютерні науки та прикладні інформаційні технології.
2. Комп'ютерна інженерія та системи захисту інформації.
3. Математичне моделювання та інженерія програмного забезпечення
4. Телерадіокомунікації, медійні та комунікаційні системи.
5. Проблеми впровадження інформаційних технологій у виробництво та управління.

Робочі мови конференції: українська, англійська

ОРГКОМІТЕТ:

СИНЮК О. М. голова оргкомітету, проректор Хмельницького національного університету з наукової роботи, доктор технічних наук, професор

САВЕНКО О. С. заступник голови оргкомітету, декан факультету Інформаційних технологій ХНУ, доктор технічних наук, професор

БАРМАК О. В. заступник голови оргкомітету, завідувач кафедри Комп'ютерних наук ХНУ, доктор технічних наук, професор

ГОВОРУЩЕНКО Т. О. завідувач кафедри Комп'ютерної інженерії та інформаційних систем ХНУ, доктор технічних наук, професор

ВИСОЦЬКА О. В. доктор технічних наук, завідувач кафедри Радіоелектронних та біомедичних комп'ютеризованих засобів і технологій Національного аерокосмічного університету ім. М. Є. Жуковського «Харківський авіаційний інститут», професор

ЛАВРОВ Є. А. доктор технічних наук, професор (Сумський державний університет)

ТИМОФЄЄВА Л. В. відповідальна за студентську науково-дослідну роботу ХНУ

МАЗУРЕЦЬ О. В. секретар конференції, к.т.н., доцент кафедри Комп'ютерних наук ХНУ

МОЛЧАНОВА М. О. секретар конференції, викладач кафедри Комп'ютерних наук ХНУ

КОНТАКТНА ІНФОРМАЦІЯ:

e-mail для листування: apkt.khnu@gmail.com

УДК 004.4

Омельяненко А.Ю., Копішинська О.П.

Полтавський державний аграрний університет

ОКРЕМІ АСПЕКТИ ВИКОРИСТАННЯ БІБЛІОТЕКИ JAVASCRIPT IMMUTABLE.JS

Розглянуто основні аспекти використання імутабельності даних як одного з принципів функціонального програмування. Показано шляхи раціоналізації роботи з кодами шляхом застосування бібліотеки Immutable.js мови програмування JavaScript. На прикладі показано ефективність саме такого підходу до вирішення завдань збереження даних.

The main aspects of using data immutability as one of the principles of functional programming are considered. The ways of streamlining the work with codes by using the library Immutable.js JavaScript programming language are shown. The example shows the effectiveness of this approach to solving data storage problems.

Сучасні вебтехнології швидко змінюються та вдосконалюються в напрямку раціоналізації роботи з програмними кодами, враховуючи досвід веброзробників у всьому світі. Серед верстальників вебсайтів та адміністраторів часто використовується бібліотека JavaScript Immutable.js [1]. Сам термін імутабельність походить від англійського Immutable, що значить незмінний. Незмінним, або імутабельним вважається об'єкт, стан якого не може бути змінено після створення. Результатом будь-якої модифікації такого об'єкта завжди буде отримано новий об'єкт, причому версія попереднього об'єкта не зміниться. Імутабельність це один із основних принципів у функціональному програмуванні, який часто застосовується у сучасних фреймворках.

Метою роботи є підбір інструментарію написання кодів для раціонального використання імутабельності даних.

У JavaScript не можна змінювати псевдо примітивні значення (числа, стрічки, null, boolean). Якщо передати дані до іншої змінної, то ця інша змінна створить нову. Стосовно об'єктів, у JavaScript вони є імутабельними. Змінні, що означають об'єкт, завжди посилаються на вихідне значення. Наприклад, нижче поданий приклад коду, який містить певний порядок звернення до змінних на чистому JavaScript, який показує нездатність імутабельності в об'єктах, що може призвести до непередбачуваної зміни даних.

```
const person1= {
  name: 'Den',
  address: 'str. Angular.22'
}
const person2= person1;
person2.address= 'str. Vue.43';
console.log(person1=== person2);
//true
console.log(person1);
//{name: 'Den', address: 'str. Vue.43'}
```

Далі – шлях розв’язання проблеми шляхом зміни коду на TypeScript, використовуючи семантику інтерфейсів ми блокуємо зміну вже ініціалізованих об’єктів [2].

```
interface Person{
  readonly name: string,
  readonly address: string,
}
const person1: Person = {
  name: 'Den',
  address: 'str. Angular.22'
}
const person2= person1;
person2.address= 'str. Vue.43';
//Cannot assign to 'address' because it is a read-only property.
```

Але використання TypeScript не найоптимальніший метод використання імутабельності, через перезапис даних, які не були зміннені. Для оптимізації зміни даних використовують бібліотеку Immutable.js.

```
import { Map } from "immutable";
const person1 = Map({ name: 'Den',
address: 'str. Angular.22' });
const person2 = person1.set('name', 'Abramov')
console.log(person1 === person2 )
//false
console.log(person1)
//Map {name: "Den", address: "str. Angular.22", constructor: Object}
console.log(person2)
//Map {name: "Abramov", address: "str. Angular.22", constructor: Object}
```

Використовуючи підхід імутабельності разом з бібліотекою Immutable.js, ми отримуємо наступні переваги.

По-перше, передбачуваність. Непоодинокі випадки, коли передані в функцію дані можуть бути випадково зіпсовані, змінені, і відстежити такі ситуації дуже складно. Мутації приховують зміни, які можуть викликати не очікуваний результат, помилки. Імутабельність спрощує архітектуру та полегшує розуміння вашого додатку. Якщо зробити усі змінні імутабельними, то можна бути упевненим, що передані значення функції не будуть змінені нею.

По-друге, витрати на пам'ять. Додавання значень до незмінного об'єкта означає створення нового екземпляра, який скопіює всі параметри існуючого об'єкта, що вартує пам'яті, але загальна структура була незмінною. Щоразу при модифікації імутабельного об'єкта створюється його копія з необхідними змінами. Це призводить до більшої витрати пам'яті, ніж під час роботи зі звичайним об'єктом. Але оскільки імутабельні об'єкти будь-коли змінюються, то можуть бути реалізовані з допомогою стратегії, що називається «загальні структури» (structural sharing). У порівнянні з вбудованими масивами та об'єктами витрата все ще буде існувати, але вона матиме фіксовану величину і зазвичай може компенсуватися іншим перевагами, доступними завдяки незмінності. Але, використовуючи бібліотеку Immutable.js та подібні можна заощадити при цих операціях. Тобто, додаючи новий елемент до масиву, ми мали би переписати всі попередні елементи + новий, але за умови використання цієї бібліотеки, буде змінено лиш декілька об'єктів.

Таким чином, наведені вище аргументи підкріплюють висновок про необхідність та переваги використання імутабельності за рахунок застосування спеціальної бібліотеки Immutable.js мови JavaScript.

Перелік посилань

1. Імутабельність в JavaScript : вебсайт. URL: <https://habr.com/ru/company/developersoft/blog/302118/>
2. Документація Immutable.js URL: <https://immutable-js.com/docs/v4.0.0>
3. Документація TypeScript URL: <https://www.typescriptlang.org/docs/handbook/interfaces.html>



АКТУАЛЬНІ ПРОБЛЕМИ КОМП'ЮТЕРНИХ НАУК 2022

ЗБІРНИК НАУКОВИХ ПРАЦЬ

Комп'ютерна верстка: **Мазурець О.В.,
Молчанова М.О.**

Підписано до друку 17.11.2022.
Версія друку «APKN2022_CorpusPaper v5mod4».

E-mail: apkt.khnu@gmail.com
ХНУ. м. Хмельницький, вул. Інститутська, 11.