

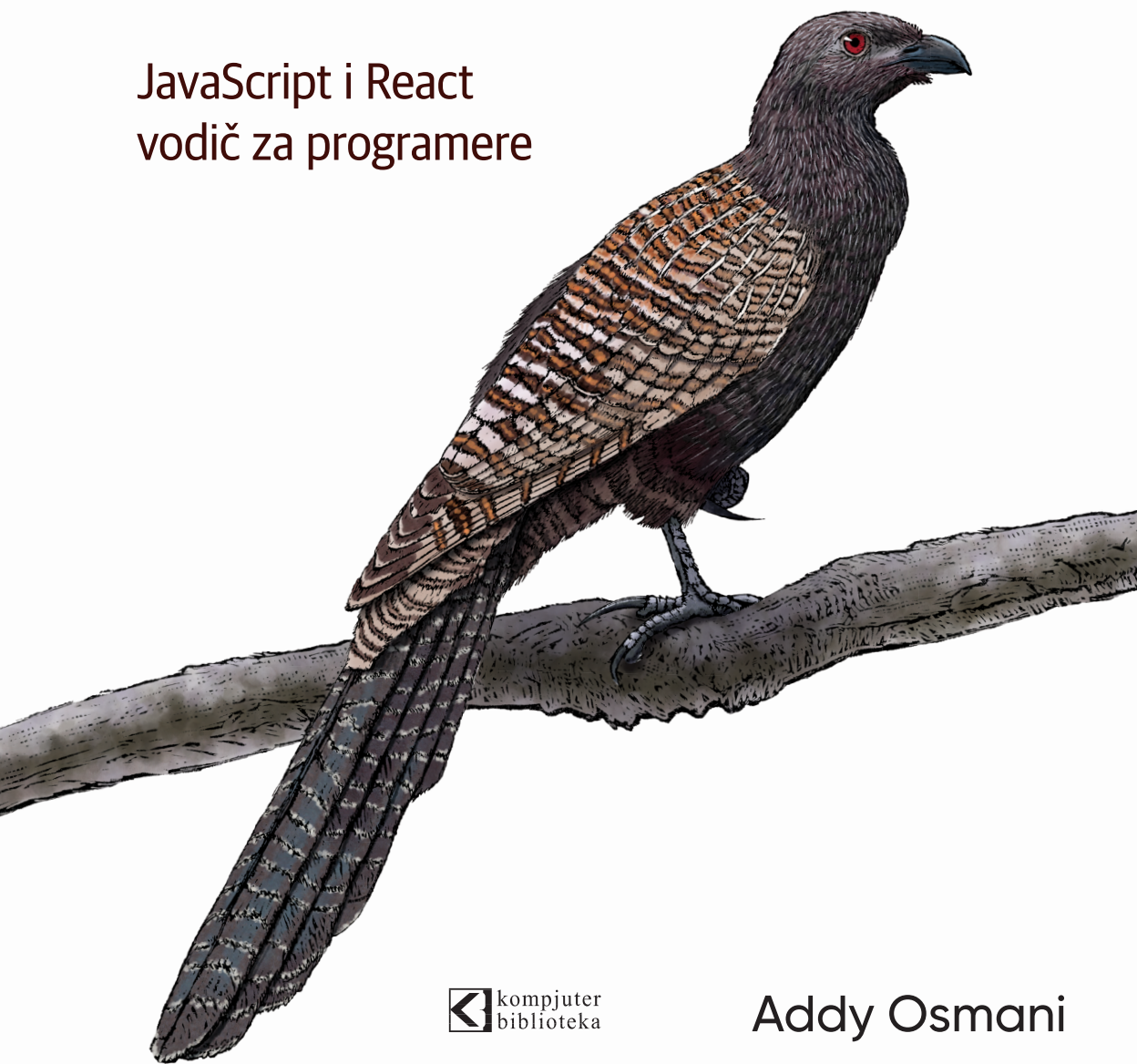
O'REILLY®

Prevod
drugog izdanja

JavaScript

PROJEKTI OBRASCI

JavaScript i React
vodič za programere



 kompjuter
biblioteka

Addy Osmani

JavaScript Projektni obrasci

Da li želite da pišete lep, uređen i lako održiv JavaScript kod primenjujući moderne projektne obrasce? Da li želite jasan i efikasan kod kojim se može upravljati? Da li želite da budete u toku sa najnovijim najboljim praksama? Ako su vaši odgovori potvrdni na sva ta pitanja, ažurirano drugo izdanje knjige JavaScript projektni obrasci je idealno mesto za početak.

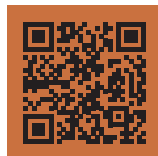
Autor Addy Osmani pokazuje kako da primenite moderne projektne obrasce u JavaScriptu i Reactu, uključujući obrasce Module, Mixin, Observer i Mediator. Učičete o obrascima performansi i prikazivanja, kao što su prikazivanje na strani servera i arhitektura ostrva. Takođe ćete naučiti kako su arhitektonski obrasci MVC, MVP i MVVM korisni iz perspektive programera moderne veb aplikacije.

U ovoj knjizi istražujemo:

- arhitektonske obrasce za uređenje vaših komponenti i aplikacija
- više od 20 projektnih obrazaca u JavaScriptu i Reactu primenljivih za programere na bilo kojem nivou
- različite kategorije obrazaca, uključujući obrasce kreiranja, strukture i ponašanja
- osnovne obrasce performansi, uključujući dinamička umetanja i deljenje koda na manje delove
- obrasce prikazivanja, kao što su prikazivanje na strani servera, tehnika hidratacije, arhitektura ostrva i još mnogo štošta

Osim toga, istražićete delove moderne sintakse JavaScript jezika, kao što su JavaScript moduli, React obrasci, kao što su Hooks, komponente višeg reda (HOC-ove) i još mnogo toga da biste napredovali u svetu veb razvoja koji prestanto evoluiraju.

Skenirajte
QR kod,
registrujte
knjigu
i osvojite
nagradu



„Addyjeva knjiga je pragmatičan pristup osnovnim konceptima koji su korisni u bezbroj namena. JavaScript projektni obrasci svojstveni su programerima koji primenjuju sistemsko razmišljanje na svoj rad. Svaki programer koji čita ovu knjigu uvideće neverovatan povratak vremena uloženog u njeno čitanje.”

– Sarah Drasner,
Direktor inženjeringa
kompanije Google

„Addy Osmani ponovo „napada“ i objavio je veoma cenjeno ažuriranje svoje već klasične knjige o projektnim obrascima u JavaScriptu.”

– Stoyan Stefanov,
WebPageTest.org inženjer i
autor knjige JavaScript obrasci

Addy Osmani je inženjer kompanije Google i vođa timova iskusnih programera koji rade na razvoju veb pregledača Chrome, pomažući tako da veb bude brz i prijatan za nadgradnju.

Prevod
drugog izdanja

JavaScript

PROJEKTNI OBRASCI

JavaScript i React vodič za programere

Addy Osmani



 kompjuter
biblioteka

O'REILLY®

Izdavač:



Obalskih radnika 4a
Beograd, Srbija

Tel: 011/2520272

e-pošta: kombib@gmail.com

veb-sajt: www.kombib.rs

Za izdavača:

Mihailo J. Šolajić, direktor

Autor:

Addy Osmani

Prevod: Biljana Tešić

Lektura: Miloš Jevtović

Recenzent: Miroslav Ristić

Slog: Zvonko Aleksić

Znak Kompjuter biblioteke:

Miloš Milosavljević

Štampa: „Pekograf“, Zemun

Tiraž: 500

Godina izdanja: 2023.

Broj knjige: 569

Izdanje: Prvo

ISBN: 978-86-7310-592-5

Naslov originala:

Learning JavaScript Design Patterns

by Addy Osmani

Copyright © 2023 Adnan Osmani. All rights reserved.

Printed in the United States of America.

Published by O'Reilly Media, Inc., 1005 Gravenstein Highway
North, Sebastopol, CA 95472.

JavaScript

Projektni obrasci

Autorizovani prevod sa engleskog jezika.

Sva prava zadržana. Nijedan deo ove knjige se ne sme reprodukovati, čuvati u sistemu za pronalaženje ili prenositi u bilo kom obliku ili na bilo koji način, bez prethodne pismene dozvole izdavača, osim u slučaju kratkih citata ugrađenih u kritičke članke ili prikaze.

Tokom pripreme ove knjige uloženi su svi napori da se obezbedi tačnost predstavljenih informacija. Međutim, informacije sadržane u ovoj knjizi se prodaju bez garancije, bilo izričite ili podrazumevane. Autori i izdavač neće biti odgovorni za bilo kakvu štetu prouzrokovanu ili navodno prouzrokovanu direktno ili indirektno ovom knjigom.

„Kompjuter biblioteka“ i „O'Reilly Media“ su nastojali da obezbede informacije o zaštitnim znakovima o svim kompanijama i proizvodima pomenutim u ovoj knjizi korišćenjem odgovarajućeg načina njihovog pominjanja u tekstu. Međutim, ne možemo da garantujemo tačnost ovih informacija.

CIP - Каталогизација у публикацији
Народна библиотека Србије, Београд

004.438JAVA
004.42:004.738.5

ОСМАНИ, Ади

JavaScript projektni obrasci : JavaScript i React vodič za programere / Addy Osmani ; prevod 2. izdanja [Biljana Tešić]. - 1. izd. - Beograd : Kompjuter Biblioteka, 2023 (Zemun : Pekograf). - XVI, 282 str. : ilustr. ; 24 cm. - (Kompjuter biblioteka ; br. knj. 569)

Prevod dela: Learning JavaScript Design Patterns. - Tiraž 500. - Registar.

ISBN 978-86-7310-592-5

a) Програмски језик „Java“
b) Интернет -- Програмирање

COBISS.SR-ID 126073353

KRATAK SADRŽAJ

PREDGOVOR	XXI
POGLAVLJE 1	
Uvod u projektne obrasce	1
POGLAVLJE 2	
Testiranje obrazaca, protoobrasci i pravilo trojke	7
POGLAVLJE 3	
Uređivanje i pisanje obrazaca	11
POGLAVLJE 4	
Antiobrasci	15
POGLAVLJE 5	
Moderna JavaScript sintaksa i karakteristike	19
POGLAVLJE 6	
Kategorije projektnih obrazaca	31
POGLAVLJE 7	
JavaScript projektni obrasci	35
POGLAVLJE 8	
JavaScript MV* obrasci	121

POGLAVLJE 9	
Obrasci asinhronog programiranja	139
POGLAVLJE 10	
Modularni JavaScript projektni obrasci	155
POGLAVLJE 11	
Obrasci Namespacing.....	175
POGLAVLJE 12	
React.js projektni obrasci	193
POGLAVLJE 13	
Obrasci Rendering.....	233
POGLAVLJE 14	
Struktura aplikacije za React.js	249
POGLAVLJE 15	
Zaključci.....	259
DODATAK	
Reference.....	261
INDEKS	

SADRŽAJ

PREDGOVOR.....	XXI
-----------------------	------------

POGLAVLJE 1

Uvod u projektne obrasce	1
Istorija projektних obrazaca	2
Šta je obrazac?	3
Slučaj svakodnevnog korišćenja projektnih obrazaca	4
Rezime	5

POGLAVLJE 2

Testiranje obrazaca, protoobrasci i pravilo trojke	7
Šta su protoobrasci?	7
Testiranje obrazaca	8
Pravilo trojke	9
Rezime	9

POGLAVLJE 3

Uređivanje i pisanje obrazaca	11
Dobro napisani obrasci	12
Pisanje obrasca	13
Rezime	14

POGLAVLJE 4

Antiobrasci	15
Šta su antiobrasci?	15
Antiobrasci u JavaScriptu	16
Rezime	17

POGLAVLJE 5

Moderna JavaScript sintaksa i karakteristike 19

Važnost razdvajanja aplikacija	19
Moduli sa ključnim rečima import i export	20
Objekti modula	22
Moduli učitani sa udaljenih izvora	23
Statički uvozi	23
Dinamički uvozi.....	23
Uvoz pri interakciji.....	24
Uvoz pri vidljivosti	24
Moduli za server.....	24
Prednosti korišćenja modula.....	25
Klase sa konstruktorima, očitavačima i postavljачima	26
Klase u JavaScript radnim okvirima	28
Rezime	29
Povezana literatura	29

POGLAVLJE 6

Kategorije projektnih obrazaca..... 31

Pogled unazad.....	31
Projektni obrasci kreiranja.....	32
Projektni obrasci strukture.....	32
Projektni obrasci ponašanja	32
Klase projektnih obrazaca	32
Rezime	34

POGLAVLJE 7

JavaScript projektni obrasci..... 35

Obrasci kreiranja	35
Obrazac Constructor	36
Kreiranje objekta.....	36
Osnovni konstruktori.....	38
Konstruktori sa prototipovima.....	39
Obrazac Module.....	40
Objektni literali	40
Obrazac Module	42
Privatnost	42
Istorija	43
Primeri	43
Varijacije obrasca Module.....	46
Uvoz umešavača.....	46
Izvozi	47
Prednosti	47
Nedostaci	48
Moderan obrazac Module sa objektom WeakMap	48
Moduli sa modernim bibliotekama.....	50

Obrazac Revealing Module.....	51
Prednosti.....	53
Nedostaci.....	53
Obrazac Singleton.....	53
Upravljanje stanjem u biblioteci React.....	57
Obrazac Prototype.....	58
Obrazac Factory.....	61
Kada se koristi obrazac Factory.....	64
Kada ne treba koristiti obrazac Factory.....	64
Obrazac Abstract Factory.....	64
Obrasci strukture.....	65
Obrazac Facade.....	66
Obrazac Mixin (Umešavač).....	68
Potklasiranje.....	68
Umešavači.....	69
Prednosti i nedostaci.....	72
Obrazac Decorator.....	73
Pseudoklasični dekorateri.....	76
Interfejsi.....	77
Apstraktni dekorateri.....	78
Prednosti i nedostaci.....	81
Flyweight.....	82
Upotreba obrazaca Flyweight.....	83
Obrasci Flyweight i deljenje podataka.....	83
Primena klasičnih obrazaca Flyweight.....	83
„Implements” za umetanje „zákpe”.....	84
Konvertovanje koda za korišćenje obrasca Flyweight.....	86
Jednostavan objekat Factory.....	89
Upravljanje spoljašnjim stanjima.....	89
Obrazac Flyweight i DOM.....	91
Primer: Centralizovano rukovanje događajima.....	91
Obrasci ponašanja.....	93
Obrazac Observer.....	93
Razlike između obrazaca Observer i Publish/Subscribe.....	97
Prednosti.....	100
Nedostaci.....	100
Implementacije obrasca Publish/Subscribe.....	100
Primer implementacije obrasca Publish/Subscribe.....	101
Korišćenje naše implementacije.....	102
Obaveštenja korisničkog interfejsa.....	103
Razdvajanje aplikacija korišćenjem Pub/Sub implementacije Bena Almana.....	105
Razdvajanje jQuery aplikacije zasnovane na Ajaxu.....	107
Obrazac Observer u React ekosistemu.....	109
Obrazac Mediator.....	110
Jednostavan Mediator.....	111
Sličnosti i razlike.....	112
Događaji.....	112

Objekti treće strane	113
Veze: kada koristiti koji obrazac	113
Upotreba obrasca Event Aggregator	114
Upotreba obrasca Mediator	114
Event Aggregator (Pub/Sub) i Mediator	115
Mediator/Middleware u modernom JavaScriptu	116
Mediator naspram Facadea	117
Obrazac Command	117
Rezime	119

POGLAVLJE 8

JavaScript MV* obrasci 121

MVC obrazac	121
Smalltalk-80 MVC obrazac	122
MVC obrazac za JavaScript programere	122
Modeli	123
Prikazi	124
Izrada šablona	126
Kontroleri	128
Šta nam omogućava MVC obrazac?	128
Smalltalk-80 MVC obrazac u JavaScriptu	128
Alternativni pogled na MVC	129
Rezime o MVC obrascu	129
MVP obrazac	129
Modeli, prikazi i prezenteri	130
MVP ili MVC obrazac?	131
MVVM obrazac	132
Istorija	132
Model	133
Prikaz	133
Model prikaza	134
Rekapitulacija: prikaz i model prikaza	134
Model prikaza naspram modela	134
Prednosti i nedostaci	135
Prednosti	135
Nedostaci	135
Poređenje MVC, MVP i MVVM obrazaca	135
Moderni MV* obrasci	136
MV* obrasci i React.js	136
Rezime	137

POGLAVLJE 9

Obrasci asinhronog programiranja 139

Asinhrono programiranje	139
Pogled unazad	141
Obrasci Promise	142
Obećanje ulančavanje	143

Obećanje rukovanje greškama	144
Obećanje paralelizam.....	144
Obećanje sekvencijalno izvršenje.....	144
Obećanje memoizacija	144
Obećanje protočnost.....	145
Obećanje pokušaj ponovo	146
Obećanje dekorater	146
Obećanje trka	147
Obrasci async/await.....	147
Async kompozicija funkcija.....	147
Async iteracija	148
Async rukovanje greškama	148
Async paralelizam.....	148
Asybc sekvencijalno izvršenje	149
Async memoizacija.....	149
Async rukovanje događajima.....	149
Async/await protočnost	150
Async pokušaj ponovo.....	150
Async/await dekorater.....	150
Dodatni praktični primeri.....	151
Kreiranje HTTP zahteva.....	151
Čitanje datoteke iz sistema datoteka	151
Pisanje u datoteku u sistemu datoteka.....	151
Izvršavanje više async operacija	152
Sekvencijalno izvršavanje više async operacija	152
Keširanje rezultata async operacije.....	152
Rukovanje događajima pomoću funkcije async/await.....	153
Ponovni pokušaj async operacije u slučaju neuspeha.....	153
Kreiranje async/await dekoratera	153
Rezime.....	154

POGLAVLJE 10

Modularni JavaScript projektni obrasci 155

Napomena o programima za učitavanje skriptova.....	155
AMD format	156
Početak upotrebe modula.....	156
AMD moduli pomoću biblioteke jQuery	160
Registrowanje biblioteke jQuery kao asinhronog modula	161
Zašto je AMD bio bolji izbor za pisanje modularnog JavaScripta?	162
Dodatna literatura za AMD	162
Programi za učitavanje skriptova i radni okviri koji podržavaju AMD	163
Zaključci o AMD formatu	163
CommonJS format	163
Početak rada	164
Upotreba višestruke zavisnosti.....	165
CommonJS u Node.js okruženju za izvršavanje	166
Da li je CommonJS pogodan za veb pregledač?.....	166
Dodatna literatura za CommonJS.....	167

I AMD i CommonJS su važeći formati modula sa različitim krajnjim ciljevima	167
UMD: AMD i CommonJS moduli kompatibilni za dodatke	168
Osnovni AMD hibridni format	168
Korišćenje CommonJS i AMD formata ili globalnih objekata veb pregledača za kreiranje modula	168
jQuery dodaci koji funkcionišu u svim okruženjima	169
Dodatna literatura za UMD i AMD	172
Rezime	173

POGLAVLJE 11

Obrasci Namespacing 175

Osnove kreiranja prostora imena	175
Pojedinačne globalne promenljive	176
Prefiks prostora imena	176
Notacija objektnih literala	177
Ugnežđeni prostori imena	180
Izrazi momentalnog pozivanja funkcije	181
Injektovanje prostora imena	184
Napredni obrasci prostora imena	186
Automatizacija ugnežđenih postora imena	186
Obrazac Dependency Declaration	188
Duboko proširenje objekta	189
Preporuka	191
Rezime	192

POGLAVLJE 12

React.js projektni obrasci 193

Uvod u React	193
Korišćena terminologija	194
Osnovni koncepti	194
Komponente višeg reda	196
Sastavljanje	199
Prednosti	199
Nedostaci	200
Obrazac Render Props	200
Podizanje stanja (lifting state)	202
Children kao funkcija	204
Prednosti	205
Nedostaci	205
Obrazac Hooks	205
Komponente klase	206
Preuređivanje	207
Kompleksnost	208
Hooks	208
Hook State	208

Hook Effect	210
Prilagođene Hook funkcije	211
Dodatni vodič za Hook funkcije.....	213
Prednosti i nedostaci upotrebe Hook funkcija.....	213
React Hookovi, naspram klasa	216
Statički uvoz	216
Dinamički uvoz.....	217
Komponente koje se mogu učitati (loadable components)	219
Uvoz pri interakciji.....	220
Uvoz pri vidljivosti	220
Deljenje koda	221
Deljenje zasnovano na putanjama.....	221
Deljenja paketa.....	222
Obrazac PRPL.....	224
Loading Prioritization	226
Učitavanje unapred u jednostraničnim aplikacijama.....	227
Učitavanje unapred + async hak.....	228
Učitavanje unapred u Chromeu 95+	228
Virtuelizacija liste	228
Kako funkcioniše windowing/virtuelizacija?	228
Liste.....	229
Mreža.....	230
Poboljšanja na veb platformi	232
Zaključci.....	232
Rezime	232

POGLAVLJE 13

Obrasci Rendering..... 233

Važnost obrazca Rendering.....	234
Prikazivanje na strani klijenta.....	236
Prikazivanje na strani servera.....	236
Statičko prikazivanje	237
Inkrementalno statičko regenerisanje	238
Inkrementalno statičko regenerisanje na zahtev.....	239
Rezime statičkog prikazivanja	239
SSR strimovanje	240
Edge SSR.....	241
Hibridno prikazivanje.....	241
Progresivna hidratacija.....	242
Hidratacija ostrva.....	243
Implementacija ostrva	243
Prednosti i nedostaci.....	245
React Server Components.....	245
Hibridno prikazivanje pomoću RSC-a i Next.js App Routera	246
Rezime	247

POGLAVLJE 14

Struktura aplikacije za React.js 249

Uvod.....	249
Grupisanje po modulu, funkciji ili putanji.....	250
Grupisanje po tipu datoteke.....	250
Hibridno grupisanje na osnovu svrhe i uobičajenih komponentata.....	251
Struktura aplikacije za moderne React funkcije.....	252
Redux.....	253
Kontejneri.....	253
Hook funkcije.....	253
Stilizovane komponente (Styled Components).....	254
Druge najbolje tehnike.....	254
Struktura aplikacije za Next.js aplikacije.....	256
Rezime.....	257

POGLAVLJE 15

Zaključci 259

DODATAK

Reference..... 261

PREDGOVOR

Svet JavaScripta je prešao dug put otkako sam napisao prvo izdanje knjige JavaScript projektni obrasci pre više od 10 godina. U to vreme sam kreirao veb aplikacije velikih razmera i otkrio da nedostatak strukture i organizacije u JavaScript kodu otežava održavanje i skaliranje tih aplikacija.

U današnje vreme „krajolik” veb razvoja se dramatično promenio. JavaScript je postao jedan od najpopularnijih programskih jezika na svetu i koristi se za sve, od jednostavnih skriptova, do složenih veb aplikacija. Jezik JavaScript je evoluirao, tako da uključuje module, obećanja i `async/await`, koji su u velikoj meri uticali na način kako dizajniramo aplikacije. Način na koji programeri pišu komponente, kao što je slučaj sa bibliotekom React, takođe je značajno uticao na to kako razmišljaju o mogućnosti održavanja. To je rezultiralo potrebom za modernim obrascima u kojima se vodi računa o tim novim izmenama.

Porastom modernih biblioteka i radnih okvira, kao što su React, Vue i Angular, programeri sada „grade” aplikacije koje su složenije nego ikada ranije. Prepoznao sam potrebu za ažuriranom verzijom knjige „JavaScript projektni obrasci”, koja bi odražavala promene u JavaScriptu i razvoju veb aplikacija.

U ovom drugom izdanju knjige cilj mi je da pomognem programerima da primene moderne projektne obrasce na njihov JavaScript kod i React aplikacije. Knjigom je obuhvaćeno više od 20 projektnih obrazaca neophodnih za izradu održivih i skalabilnih aplikacija. Knjiga nije posvećena samo projektnim obrascima, već i obrascima prikazivanja i performansi, koji su ključni za uspeh modernih veb aplikacija.

U prvom izdanju ove knjige fokusirali smo se na klasične projektne obrasce, kao što su obrasci Module, Observer i Mediator. Ti obrasci su i danas važni i relevantni, ali svet veb razvoja je značajno evoluirao u protekloj deceniji i pojavili su se novi obrasci.

Ovim novim izdanjem obuhvaćeni su ti novi obrasci, kao što su obećanja, `async/await` i novije varijacije obrasca Module. Takođe su obuhvaćeni arhitektonski obrasci, kao što su MVC, MVP i MVVM, i razmatrano gde se moderni radni okviri uklapaju u te arhitektonske obrasce.

Današnji programeri su izloženi mnogim projektnim obrascima specifičnim za biblioteku ili radni okvir. Reactov „zreli” ekosistem i korišćenje novijih JavaScript primitiva obezbeđuju odličnu lansirnu platformu za razgovor o najboljim tehnikama i obrascima u kontekstu radnog okvira ili biblioteke. Osim klasičnih projektnih obrazaca, ovom knjigom su obuhvaćeni moderni React obrasci, kao što su Hook funkcije, komponente višeg reda (Higher-Order Components) i render props objekti (Render Props). Ti obrasci su specifični za React i neophodni su za kreiranje modernih veb aplikacija pomoću popularnih radnih okvira.

Ova knjiga nije samo o obrascima, već i o najboljim tehnikama. Razmotrene su teme kao što su organizacija koda, performanse i prikazivanje, koje su ključne za kreiranje visokokvalitetnih veb aplikacija. Učićete o dinamičkim uvozima, deljenju koda, prikazivanju na strani servera, hidrataciji i arhitekturi ostrva, a to je sve od suštinskog značaja za izradu brzih i prilagodljivih veb aplikacija.

Nakon što pročitate celu knjigu, dobro ćete razumeti projektne obrasce i načine na koje možete da ih primenjujete na svoj JavaScript kod i React aplikacije. Takođe ćete znati koji obrasci su relevantni za moderni veb, a koji nisu. Ova knjiga nije samo referenca za obrasce, već je i vodič za izradu visokokvalitetnih veb aplikacija. Naučićete kako da uređujete svoj kod za maksimalnu mogućnost održavanja i skalabilnost i kako da optimizujete svoj kod za performanse.

Struktura knjige

Ova knjiga je organizovana u 15 poglavlja, koja su osmišljena tako da vas provedu kroz JavaScript projektne obrasce iz moderne perspektive, uključujući ažurirane jezičke funkcije i obrasce specifične za React. Svako poglavlje se nadovezuje na prethodno, što omogućava da postepeno razvijate svoje znanje i da ga efikasno primenjujete:

- Poglavlje 1, „Uvod u projektne obrasce”: upoznajte istoriju projektnih obrazaca i njihov značaj u svetu programiranja
- Poglavlje 2, „Testiranje obrazaca, protoobrasci i pravilo trojke”: razumevanje procesa evaluacije i fino podešavanje projektnih obrazaca
- Poglavlje 3, „Uređivanje i pisanja obrazaca”: naučite „anatomiju” dobro napisanog obrasca i kako ga možete kreirati
- Poglavlje 4, „Antiobrasci”: otkrijte šta su antiobrasci i kako da ih izbegnete u svom kodu
- Poglavlje 5, „Moderna JavaScript sintaksa i karakteristike”: istražite najnovije karakteristike JavaScript jezika i njihov uticaj na projektne obrasce
- Poglavlje 6, „Kategorije projektnih obrazaca”: „zaronite” u različite kategorije projektnih obrazaca: kreiranja, strukture i ponašanja
- Poglavlje 7, „JavaScript projektni obrasci”: proučite više od 20 klasičnih projektnih obrazaca u JavaScriptu i njihove moderne adaptacije
- Poglavlje 8, „JavaScript MV* obrasci”: saznajte više o arhitektonskim obrascima, kao što su MVC, MVP i MVVM, i o njihovom značaju u modernom veb razvoju

- ▣ Poglavlje 9, „Obrasci asinhronog programiranja”: razumevanje moći asinhronog programiranja u JavaScriptu i različitim obrazaca za rukovanje njime
- ▣ Poglavlje 10, „Modularni JavaScript projektni obrasci”: otkrijte obrasce za organizovanje i modularizaciju vašeg JavaScript koda
- ▣ Poglavlje 11, „Obrasci Namespacing”: naučite različite tehnike za postavljanje prostora imena vašeg JavaScript koda da biste izbegli globalno zagađivanje prostora imena
- ▣ Poglavlje 12, „React.js projektni obrasci”: istražite obrasce specifične za React, uključujući komponente višeg reda, props objekte za prikazivanje i Hook funkcije
- ▣ Poglavlje 13, „Obrasci Rendering”: razumevanje različitih tehnika prikazivanja, kao što su prikazivanje na strani klijenta, prikazivanje na strani servera, progresivna hidratacija i arhitektura ostrva
- ▣ Poglavlje 14, „Struktura aplikacije za React.js”: naučite kako da strukturirate svoju React aplikaciju za bolju organizaciju, održivost i skalabilnost
- ▣ Poglavlje 15, „Zaključci”: završavamo knjigu važnim zaključcima i završnim mislima

U knjizi se nalaze praktični primeri u kojima su ilustrovani obrasci i koncepti koji se razmatraju. Nakon vašeg „putovanja”, dobro ćete razumeti JavaScript projektne obrasce i bićete spremni za pisanje elegantnog, održivog i skalabilnog koda.

Bez obzira da li ste iskusni veb programer ili početnik, ova knjiga će vam obezbediti znanje i alatke koji su vam potrebni za izradu modernih, održivih i skalabilnih veb aplikacija. Nadam se da će vam ova knjiga biti dragocen resurs kada budete nastavljali da razvijate svoje veštine i izrađujete neverovatne veb aplikacije.

Konvencije upotrebljene u ovoj knjizi

U ovoj knjizi su upotrebljene sledeće tipografske konvencije:

Italic

Ukazuje na nove termine, URL adrese, adrese e-pošte, nazive datoteka i ekstenzije datoteka.

`Constant width`

Upotrebljen za listinge programa, kao i unutar pasusa za ukazivanje na elemente programa, kao što su nazivi promenljivih ili funkcija, na baze podataka, na tipove podataka, na promenljive okruženja, na iskaze i na ključne reči.

Constant width italic

Prikazuje tekst koji treba zameniti vrednostima koje je uneo korisnik ili vrednostima koje su određene kontekstom.



Ovaj element označava savet ili predlog.



Ovaj element označava opštu napomenu.

Upotreba primera koda

Ako imate tehničko pitanje ili problem u vezi sa upotrebom primera koda, molimo vas da pošaljete e-mail na adresu bookquestions@oreilly.com.

Ova knjiga će vam pomoći da obavite svoj posao. Generalno, ako je primer koda obezbeđen za knjigu, možete da ga upotrebite u vašim programima i dokumentaciji. Ne treba da kontaktirate sa nama da biste tražili dozvolu, osim ako reprodukujete značajan deo koda. Na primer, pisanje programa koji koristi nekoliko blokova koda iz ove knjige ne zahteva dozvolu. Prodaja i distribuiranje primera iz knjiga izdavača O'Reilly zahteva dozvolu. Odgovaranje na pitanja citiranjem ove knjige i primera koda ne zahteva dozvolu. Uključivanje značajnog dela primera koda iz ove knjige u dokumentaciji vašeg proizvoda zahteva dozvolu.

Cenimo, ali generalno ne zahtevamo citiranje. Referenca obično uključuje naslov, autora, izdavača i ISBN. Na primer, „Learning JavaScript Design Patterns, 2nd ed., by Addy Osmani (O'Reilly). Copyright 2023 Adnan Osmani, 978-1-098-13987-2”.

Ako smatrate da vaša upotreba primera koda spada van fer upotrebe ili prethodno opisanih dozvola, slobodno nam se javite na adresu permissions@oreilly.com.



Postanite član Kompjuter biblioteke

Kupovinom jedne naše knjige stekli ste pravo da postanete član Kompjuter biblioteke. Kao član možete da kupujete knjige u pretplati sa 40% popusta i učestvujete u akcijama kada stvarujete popuste na sva naša izdanja. Potrebno je samo da se prijavite preko formulara na našem sajtu.
Link za prijavu: kombib.rs/kblista.php

Skenirajte QR kod
registrujte knjigu
i osvojite nagradu



POGLAVLJE

1

Uvod u projektne obrasce

Dobar kod je kao ljubavno pismo za sledećeg programera koji će ga održavati!

Projektni obrasce obezbeđuju zajednički rečnik za uređenje koda, čineći ga lakšim za razumevanje. Oni pomažu drugim programerima da se poboljša kvalitet te veze. Poznavanje projektnih obrazaca nam pomaže da identifikujemo teme koje se ponavljaju u zahtevima i da ih mapiramo u konačna rešenja. Možemo se osloniti na iskustvo drugih koji su se susreli sa sličnim problemima i osmislili optimizovani metod za njihovo rešavanje. To znanje je neprocenjivo, jer ono utire put za pisanje ili prepravljavanje koda koji se može lako održavati.

Bilo na strani servera ili strani klijenta, JavaScript je „kamen-temeljac” modernog razvoja veb aplikacija. Prethodno izdanje ove knjige fokusirano je na nekoliko popularnih projektnih obrazaca u kontekstu JavaScripta. Tokom godina, sintaksa JavaScript jezika i njegove mogućnosti značajno su napredovale. Sada JavaScript podržava module, klase, lambda funkcije i šablonske literale, koje ranije nije podržavao. Takođe imamo napredne JavaScript biblioteke i radne okvire koji su olakšali život mnogim veb programerima. Koliko su onda relevantni projektne obrasce u kontekstu modernog JavaScripta ?

Važno je napomenuti da projektne obrasce tradicionalno nisu propisani, niti specifični za jezik. Možete ih primeniti kada mislite da odgovaraju, ali i ne morate. Baš kao strukture podataka ili algoritme, još uvek možete da primenite klasične projektne obrasce, koristeći moderne programske jezike, uključujući i JavaScript. Možda vam neće biti potrebni neki od tih projektnih obrazaca u modernim radnim okvirima ili bibliotekama u kojima su već apstrahovani. Nasuprot tome, upotreba specifičnih obrazaca može čak biti podstaknuta nekim radnim okvirima.

U ovom izdanju primenjujemo pragmatičan pristup obrascima. Istražićemo zašto određeni obrasce mogu biti prikladni za implementaciju određenih funkcija i da li se neki obrazac i dalje preporučuje u kontekstu modernog JavaScripta.

Pošto su aplikacije postale interaktivnije i zahtevaju veliku količinu JavaScript koda, JavaScript jezik se našao na udaru stalnih kritika zbog svog negativnog uticaja na performanse. Programeri neprestano traže nove obrasce koji mogu da optimizuju performanse JavaScripta. U ovom izdanju ističemo takva poboljšanja gde god je to relevantno. Takođe ćemo razmatrati obrasce specifične za radne okvire, kao što su React obrasci Hooks i komponente višeg reda, koje su postale sve popularnije u doba biblioteke React.js.

Vraćajući se korak unazad, prvo ćemo istražiti istoriju i značaj projektnih obrazaca. Ako vam je već poznata ta istorija, za nastavak čitanja slobodno pređite na odeljak „Šta je obrazac?” na strani 2.

Istorija projektnih obrazaca

Projektni obrasci su potekli od ranog rada arhitekta Christophera Alexandra, koji je često pisao o svojim iskustvima u rešavanju problema dizajna i o tome kako se ti problemi odnose na zgrade i gradove. Jednog dana je shvatio da određene konstrukcije dizajna dovode do željenog optimalnog efekta kada se koriste više puta.

Christopher Alexander je osmislio jezik obrazaca, u saradnji sa još dvoje arhitekata - sa Sarom Ishikawa i Murrayom Silversteinom. Taj jezik omogućava da svako ko želi dizajnirati i gradi na bilo kojem nivou. Objavljen je 1977. godine u radu „Jezik obrazaca”, koji je kasnije objavljen kao knjiga.

Oko 1990. godine softverski inženjeri su počeli da uključuju principe, o kojima je Alexander pisao, u prvu dokumentaciju o projektnim obrascima kako bi vodili programere početnike koji žele da poboljšaju svoje veštine programiranja. Važno je napomenuti da su koncepti koji stoje iza projektnih obrazaca prisutni u IT industriji od njenog početka, doduše u manje formalizovanom obliku.

Jedan od prvih i verovatno najpoznatijih objavljenih zvaničnih radova o projektnim obrascima u softverskom inženjerstvu bila je knjiga iz 1995. godine Design Patterns: Elements of Reusable Object-Oriented Software, koju su napisali Erich Gamma, Richard Helm, Ralph Johnson i John Vlissides. Većina inženjera danas tu grupu prepoznaje kao Veliku četvorku (Gang of Four - GoF).

Knjiga Velike četvorke je bila posebno korisna u unapređenju koncepta projektnih obrazaca u oblasti razvoja softvera. U njoj je opisano nekoliko razvojnih tehnika i „zamki” i obezbeđena su 23 osnovna objektno-orijentisana projektna obrasca, koja se danas često koriste širom sveta. Te obrasce ćemo detaljnije razmotriti u poglavlju 6, a oni takođe čine osnovu za naše razmatranje u poglavlju 7.

Šta je obrazac?

Obrazac je šablon rešenja za višekratnu upotrebu koji možete primeniti na probleme i teme koji se ponavljaju u dizajnu softvera. Baš kao i u drugim programskim jezicima, kada izrađujete JavaScript veb aplikaciju, možete koristiti šablon da uredite svoj JavaScript kod u različitim situacijama kada mislite da će vam to pomoći.

Učenje i korišćenje projektnih obrazaca je uglavnom korisno za programere zbog sledećih stavki:

Obrasci su dokazana rešenja.

Obrasci su rezultat kombinovanog iskustva i shvatanja programera koji su pomogli u njihovom definisanju. To su provereni pristupi za koje se zna da funkcionišu pri rešavanju specifičnih problema u razvoju softvera.

Obrasci se mogu lako ponovo upotrebiti.

Obrazac obično obezbeđuje rešenje koje možete da usvojite i prilagodite svojim potrebama. Ta karakteristika čini obrasce prilično robusnim.

Obrasci mogu biti izražajni.

Obrasci mogu pomoći da se izraze elegantna rešenja za opsežne probleme korišćenjem postavljene strukture i deljenog rečnika.

Ovo su dodatne prednosti koje obrasci obezbeđuju:

Obrasci pomažu u sprečavanju manjih problema koji mogu izazvati druge značajne probleme u procesu razvoja aplikacije.

Kada koristite utvrđene obrasce za kreiranje koda, ne morati brinuti da li ćete dobiti pogrešnu strukturu i možete se fokusirati na kvalitet celokupnog rešenja. Obrazac vas podstiče da pišete uređeniji i organizovaniji kod na prirodan način, izbegavajući potrebu za prepravljajanjem tog koda u budućnosti radi jasnoće.

Obrasci obezbeđuju generalizovana rešenja koja su dokumentovana na način koji ne zahteva da budu u vezi sa određenim problemom.

Taj generalizovani pristup znači da možete primeniti projektne obrasce da biste poboljšali strukturu koda, bez obzira na aplikaciju (i u mnogim slučajevima bez obzira na programski jezik).

Neki obrasci mogu smanjiti ukupnu veličinu koda izbegavanjem ponavljanja.

Projektni obrasci podstiču programere da bolje pogledaju svoja rešenja za oblasti u kojima mogu da postignu trenutno smanjenje dupliranja. Na primer, možete smanjiti broj funkcija koje izvršavaju slične procese na jednu generalizovanu funkciju da biste smanjili vaš izvorni kod.

Obrasci doprinose rečniku programera, što čini komunikaciju bržom.

Programeri mogu spomenuti obrazac kada komuniciraju sa svojim timom, kada o obrascu diskutuju u zajednici dizajnera i korisnika projektnih obrazaca ili, indirektno, kada drugi programer kasnije održava kod.

Popularni projektni obrasci mogu se dodatno improvizovati iskorišćavanjem zajedničkih iskustava programera koji koriste te obrasce i doprinosom zajednici.

U nekim slučajevima to dovodi do kreiranja potpuno novih projektnih obrazaca, dok u drugim može dovesti do poboljšanih smernica za korišćenje specifičnih obrazaca. To može osigurati da rešenja zasnovana na obrascima i dalje budu robusnija od ad hoc rešenja.



Obrasci *nisu* konkretna rešenja. Uloga obrasca je samo da nam obezbedi šemu rešenja. Obrasci ne rešavaju sve probleme dizajna, niti zamenjuju dobre dizajnere softvera. I dalje su vam potrebni dobri dizajneri da odaberete odgovarajuće obrasce koji mogu poboljšati celokupni dizajn.

Slučaj svakodnevnog korišćenja projektnih obrazaca

Ako ste koristili React.js, verovatno ste naišli na obrazac Provider. Ako niste, možda ste iskusili sledeću situaciju.

Stablu komponentata u veb aplikacijama često je potreban pristup deljenim podacima, kao što su korisničke informacije ili korisničke dozvole za pristup. Tradicionalni način da se pristupi tim podacima u JavaScriptu je postavljanje tih svojstava za korenu komponentu, a zatim njihovo prosleđivanje sa roditeljske na dete-komponente. Pošto se hijerarhija komponentata produbljuje i postaje sve više ugnežđena, vi je analizirate pomoću svojih podataka, što rezultira praksom dubljeg sagledavanja props objekata unutar komponenti. To dovodi do neodrživog koda u kojem će se podešavanje svojstava i prosleđivanje ponavljati u svakoj dete-komponenti koja se oslanja na te podatke.

React i nekoliko drugih radnih okvira rešavaju taj problem pomoću obrasca Provider. Pomoću obrasca Provider, React Context API može emitovati stanje/podatke na više komponentata pomoću provajdera konteksta (contextprovider). Dete-komponente kojima su potrebni deljeni podaci mogu da pristupe tom provajderu kao „potrošači” konteksta ili da koriste `useContext` Hook.

Obrazac Provider je odličan primer projektnog obrasca koji se koristi za optimizaciju rešenja nekog uobičajenog problema. U ovoj knjizi ćemo detaljno razmotriti taj i mnoge takve obrasce.

Rezime

Uz uvodno razmatranje važnosti projektnih obrazaca i njihove relevantnosti za moderni JavaScript, sada možemo duboko „zaroniti” u učenje JavaScript projektnih obrazaca. U prvih nekoliko poglavlja u ovoj knjizi razmatramo uređivanje i klasifikaciju obrazaca i identifikaciju antišablona pre nego što ćemo da razmotrimo specifičnosti projektnih obrazaca za JavaScript. Prvo ćemo u sledećem poglavlju da vidimo šta je potrebno da bi predloženi „protoobrazac” bio prepoznat kao obrazac.

Testiranje obrazaca, protoobrasci i pravilo trojke

Od trenutka kada je novi obrazac predložen do njegovog potencijalnog širokog usvajanja možda će dizajnerska zajednica i softverski programeri morati da izvrše višestruke detaljne provere obrasca. U ovom poglavlju razmatramo „putovanje” upravo uvedenog protoobrasca kroz postupak testiranja obrasca, sve dok na kraju ne bude prepoznat kao obrazac ukoliko ispunjava *pravilo trojke*.

U ovom i sledećem poglavlju istražujemo pristup uređivanju, pisanju, predstavljanju i recenziranju projektnih obrazaca u nastajanju. Ako želite da prvo učite o ustaljenim projektnim obrascima, za sada možete preskočiti ova dva poglavlja.

Šta su protoobrasci?

Svaki algoritam, najbolja praksa ili rešenje ne predstavljaju ono što bi se moglo smatrati kompletnim obrascem. Možda nedostaje nekoliko ključnih „sastojaka”, ali i dizajnerska zajednica je generalno oprezna prema nečemu za šta se tvrdi da je obrazac bez opsežne i kritičke evaluacije. Čak i ako nam je predstavljeno nešto što *izgleda kao* da ispunjava kriterijume za neki obrazac, ne bi trebalo da ga smatramo obrascem, sve dok dizajnerska zajednica ne izvrši odgovarajuća ispitivanja i testiranja.

U svom radu, Alexander tvrdi da obrazac treba da bude i proces i „stvar”. Ta definicija je besmislena, jer Alexander se nadovezuje na nju, ističući da proces treba da kreira „stvar”. Zbog toga se obrasci, uglavnom, fokusiraju na bavljenje vizuelno prepoznatljivom strukturom; trebalo bi da možemo da vizuelno prikažemo (ili nacrtamo) sliku koja predstavlja strukturu koja je rezultat primene obrasca u praksi.

Testiranje obrazaca

Često možete naići na termin protoobrazac kada proučavate projektne obrasce. Šta je protoobrazac? Dakle, obrazac koji još uvek nije konačno „prošao” testiranje obrazaca obično se naziva protoobrazac. Protoobrasci mogu biti rezultat rada nekoga ko je utvrdio određeno rešenje koje je vredno deljenja sa zajednicom. Međutim, pošto je predloženo rešenje relativno novo, zajednica nije imala priliku da ga na odgovarajući način ispita.

Alternativno, pojedinci koji dele obrazac možda neće imati vremena ili interesovanja da sprovedu proces testiranja obrasca i, umesto toga, mogu objaviti kratak opis svog protoobrasca. Kratki opisi ili isecci koda obrasca te vrste poznati su kao *patleti*.

Posao uključen u sveobuhvatno dokumentovanje kvalifikovanog obrasca može biti prilično zastrašujući. Ako se osvrnemo na neke od najranijih radova u oblasti projektnih obrazaca, videćemo da se obrazac može smatrati „dobrim” ako:

rešava određeni problem

Obrasci ne bi trebalo da obuhvataju samo principe ili strategije. Oni moraju da pronađu rešenja. To je jedan od najvažnijih „sastojaka” dobrog obrasca.

nema očigledno rešenje

Možemo otkriti da tehnike rešavanja problema često proizilaze iz dobro poznatih osnovnih načela. Najbolji projektni obrasci obično obezbeđuju rešenja za probleme indirektno – to se smatra neophodnim pristupom za najizazovnije probleme povezane sa dizajnom.

opisuje dokazani koncept

Projektni obrasci zahtevaju dokaz da funkcionišu onako kako je opisano; bez tog dokaza, dizajn se ne može ozbiljno razmatrati. Ako je obrazac po svojoj prirodi nesiguran, samo hrabri će pokušati da ga upotrebe.

opisuje vezu

U nekim slučajevima može izgledati da obrazac opisuje tip modula. Bez obzira kako implementacija izgleda, u zvaničnom opisu obrasca moraju da budu opisane mnogo dublje sistemske strukture i mehanizmi koji objašnjavaju njegovu vezu sa kodom.

Biće vam oprošteno ako mislite da protoobrazac koji ne ispunjava smernice nije vredan učenja; međutim, to je daleko od istine. Mnogi protoobrasci su, zapravo, prilično dobri. Ne kažem da je vredno razmatrati sve protoobrasce, ali postoji dosta korisnih protoobrazaca koji bi nam mogli pomoći u budućim projektima. Najbolje prosudite sami, imajući na umu prethodnu listu, i izabraćete dobre protoobrasce.

Pravilo trojke

Jedan od dodatnih uslova da bi obrazac bio validan je da prikazuje neki fenomen koji se ponavlja. To često možete podeliti na najmanje tri ključne oblasti, koje se nazivaju pravilo trojke. Da biste prikazali ponavljanje korišćenjem tog pravila, morate pokazati sledeće:

Pogodnost za upotrebu

Kako se obrazac smatra uspešnim?

Korisnost

Zašto se obrazac smatra uspešnim?

Primenljivost

Da li je dizajn vredan da bude obrazac jer ima širu primenljivost? Ako jeste, to treba objasniti. Kada pregledate ili definišete obrazac, važno je da imate na umu te oblasti.

Rezime

U ovom poglavlju je pokazano da svaki predloženi protoobrazac ne mora uvek biti prihvaćen kao obrazac. U sledećem poglavlju su prikazani osnovni elementi i najbolje tehnike za uređivanje i dokumentovanje obrazaca, tako da ih zajednica može lako razumeti i iskoristiti.