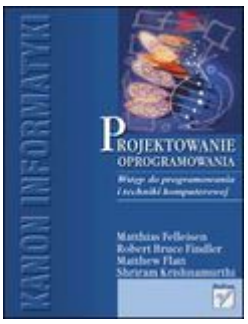

Projektowanie oprogramowania. Wstęp do programowania i techniki komputerowej

Matthias Felleisen, Robert Bruce Findler, Matthew Flatt, Shriram Krishnamurthi

Drogi Czytelniku! Poniżej zamieszczona jest errata do książki: **"Projektowanie oprogramowania. Wstęp do programowania i techniki komputerowej"**

Jest to lista błędów znalezionych po opublikowaniu książki, zgłoszonych i zaakceptowanych przez naszą redakcję. Pragniemy, aby nasze publikacje były wiarygodne i spełniały Twoje oczekiwania. Zapoznaj się z poniższą listą. Jeśli masz dodatkowe zastrzeżenia, możesz je zgłosić pod adresem https://helion.pl/user/erraty

Strona	Linia	Jest	Powinno
42	16	(* (/ 50 0.10) (- 20 cena-biletu))))	(* (/ 10 0.50) (- 20 cena-biletu))))
45	3	wysokość " * v * t,	wysokość 1/2 * v * t,
56	3 od dołu	Jeśli to możliwe, przekształć te przykłady na wyrażenia logiczne.	Pamiętaj, że przykłady mają teraz postać wyrażeń logicznych.
59	12 od dołu	Możesz na razie zostawić ... w miejscu pozostałych odpowiedzi.	Możesz na razie zostawić ... w miejscu na pozostałe odpowiedzi.
60	8 od dołu	przekraczającego 3700 złotych brutto	przekraczającego 37000 złotych brutto
61	11	otrzyma 24 złote,	otrzyma 25 złotych
65	11 od dołu	(odpowiedz ´JakSie Masz?).	(odpowiedz ´JakSieMasz?).

75	12 od dołu	Stworzysz w ten sposób sygnalizację z samymi czerwonymi światłami.	Stworzysz w ten sposób sygnalizację z jednym czerwonym światłem.
90	17 od dołu	Piksel znajduje się wewnątrz okręgu,	Piksel znajduje się wewnątrz prostokąta,
114	18	pierwszy element definicji nazywany jest funkcją	pierwszy element wywołania nazywany jest funkcją
116	19	false: jeśli pierwszy warunek jest prawdziwy	true: jeśli pierwszy warunek jest prawdziwy
116	9	true: jeśli pierwszy warunek jest fałszywy	false: jeśli pierwszy warunek jest fałszywy
123	12	... czyli 3,14 ...	... czyli 314,0 ...
131	8 od dołu. Podobny błąd także na stronie 132	dodaje trzy liczby do listy	dodaje trzy liczby z listy
133	11	(define (nasz-cons dana-wartosc dana-lista)(make-para dana-wartosc dana-lista))	(define (nasz-cons dana-wartosc dana-lista)
133	8 od dołu	[(nasz-cons? dana-lista) (make-pair dowolny-element dana-lista)]	[(nasz-cons? dana-lista) (make-pair dana-wartosc dana-lista)]
133	9 od dołu	[(empty? dana-lista)(make-para dowolny-element dana-lista)]	[(empty? dana-lista)(make-para dana-wartosc dana-lista)]
137	10	... i (first dana-lista-symboli):	... i (rest dana-lista-symboli):
153	8	cons (make-es ´lalka 17.95) empty)	(cons (make-es ´lalka 17.95) empty)

162	14 od dołu	(dodaj1 0)	(add1 0)
162	17	(dodaj1 (dodaj1 (dodaj1 0)))	(add1 (add1 (add1 0)))
162	18	(dodaj1 (dodaj1 (dodaj1 (dodaj1 0))))	(add1 (add1 (add1 (add1 0))))
162	7 od dołu	(odejmij1 (dodaj1 n))=n	(sub1 (add1 n))=n
164	6 od dołu	(cons ′hello empty)	(cons ′witaj empty)
165	4, 7, 8	(powitania (odejmij1 n))	(powitania (sub1 n))
168	5 od dołu	(! (odejmij1 n))	(! (sub1 n))
171	20 od dołu	(dodaj1 min), (dodaj1 (dodaj1 min)),	(add1 min), (add1 (add1 min)),
174	15 od dołu	(dodaj1 (dodaj-do-pi	(add1 (dodaj-do-pi
174	6, 14, 16, 19 od dołu (podobne błędy na stronie 172)	(odejmij1 n)	(sub1 n)
182	1 od dołu, oraz w dalszej części przykładu	(define-struct list (od data wiadomosc))	(define-struct mail (od data wiadomosc))
183	2 od dołu	(make-list nazwisko n s)	(make-mail nazwisko n s)
202	5	(make-dziecko empty empty ′Carl 1926 ′green)	(make-dziecko empty empty ′Karol 1926 ′zielone)
203	15 od dołu	[else ;(dziecko? dane-drzewo-genealogiczne)	[else (dziecko? dane-drzewo-genealogiczne)