



# Siedem języków w siedem tygodni. Praktyczny przewodnik nauki języków programowania

Bruce A. Tate

Drogi Czytelniku! Poniżej zamieszczona jest errata do książki:

## "Siedem języków w siedem tygodni. Praktyczny przewodnik nauki języków programowania"

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
217	33	9 > X < 0 -> negative;	9 > X < 0 -> fałsz;
269	23	repl odpowiedziała, zwracając sekwencję.	rest odpowiedziała, zwracając sekwencję.

Poniżej znajduje się lista błędów znalezionych przez czytelników, ale jeszcze nie potwierdzonych przez Redakcję:

Strona	Linia	Jest	Powinno
46	-	<b>drzewo_ruby.odwiedź = { węzeł  puts węzeł.nazwa_węzła}</b>	<b>drzewo_ruby.odwiedź { węzeł  puts węzeł.nazwa_węzła}</b>
46	-	<b>drzewo_ruby.odwiedź_wszystkie = { węzeł  puts węzeł.nazwa_węzła}</b>	<b>drzewo_ruby.odwiedź_wszystkie { węzeł  puts węzeł.nazwa_węzła}</b>
48	-	<b>Moduł jest kolekcją funkcji i stałych. Jeśli włączymy model do klasy, [...]</b>	<b>Moduł jest kolekcją funkcji i stałych. Jeśli włączymy moduł do klasy, [...]</b>

74	10	<b>lo&gt; metoda() type</b>	<b>list(1, 2, 3, 4)</b>
120	22	<b>Druga klauzula jest bardziej złożona: przodek(X, Y) :- ojciec(X, Z), ojciec(Z, Y).</b>	<b>Druga klauzula jest bardziej złożona: przodek(X, Y) :- ojciec(X, Z), przodek(Z, Y).</b>
158	9	<b>batate\$ scala code/scala/forLoop.scala wszystko całość składa się z kawałków</b>	<b>batate\$ scala code/scala/forLoop.scala całość składa się z kawałków</b>
161	26	<b>println("Obrót " + zmieńKierunek + "Obecne nastawy " + direction)</b>	<b>println("Obrót " + zmieńKierunek + "Obecne nastawy " + kierunek)</b>
169	30	<b>scala&gt; var zmienna_niemutowalna = "Nie jestem mutowalna"</b>	<b>scala&gt; val zmienna_niemutowalna = "Nie jestem mutowalna"</b>
184	1	<b>scala&gt; movies.text</b>	<b>scala&gt; filmy.text</b>
206	19	<b>4 + "ciąg_znaków".</b>	<b>4 + "ciag_znakow".</b>
206	4	<b>"ciąg znaków".</b>	<b>"ciag znakow".</b>
207	14	<b>Pigułka = niebieska.</b>	<b>Pigulka = niebieska.</b>
208	25	<b>Osoba = {osoba, {nazwisko, "Agent Smith"}, {zawód, "Zabijanie programów"} }.</b>	<b>Osoba = {osoba, {nazwisko, "Agent Smith"}, {zawod, "Zabijanie programow"} }.</b>

208	8	{komiks, {nazwa, "Calvin i Hobbes"}}, {postać, "Spaceman Spiff"}}.	{komiks, {nazwa, "Calvin i Hobbes"}}, {postać, "Spaceman Spiff"}}.
209	13	[Głowa   Ogon] = [1, 2, 3].	[Głowa   Ogon] = [1, 2, 3].
209	15	Głowa.	Głowa.
212		lustro(śmiejąca_się_gęba).	lustro(smiejaca_sie_geba).
212		prosty:lustro(śmiejąca_się_gęba).	prosty:lustro(smiejaca_sie_geba).
216	13	Zwierzę= "pies".	Zwierze = "pies".
216	14	case Zwierzę of	case Zwierze of
216	23	case Zwierzę of "słoń" -> dumbo; _ -> coś_innego end.	case Zwierze of "slon" -> dumbo; _ -> cos_innego end.
217	20	X = 0. if X > 0 -> prawda; X < 0 -> fałsz end.	X = 0. if X > 0 -> prawda; X < 0 -> fałsz end.
217	31	if X > 0 -> prawda; X < 0 -> negative; true -> zero end.	if X > 0 -> prawda; X < 0 -> nieprawda; true -> zero end.
220	3	Wyświetl = fun(X) -> io:format("~p~n", [X]) end.	Wyswietl = fun(X) -> io:format("~p~n", [X]) end.

222	16	<b>Sumator =</b> <b>fun(ElementListy,</b> <b>SumaCzęściowa) -</b> <b>&gt; ElementListy +</b> <b>SumaCzęściowa</b> <b>end.</b>	<b>Sumator =</b> <b>fun(ElementListy,</b> <b>SumaCzesciowa) -</b> <b>&gt; ElementListy +</b> <b>SumaCzesciowa</b> <b>end.</b>
222	18	<b>SumaPoczątkowa</b> <b>= 0.</b> <b>0</b> <b>lists:foldl(Sumator,</b> <b>SumaPoczątkowa,</b> <b>Liczby).</b>	<b>SumaPoczątkowa</b> <b>= 0.</b> <b>0</b> <b>lists:foldl(Sumator,</b> <b>SumaPoczątkowa,</b> <b>Liczby).</b>
225	14	<b>ZVAT =</b> <b>[{Produkt, Ilość,</b> <b>Cena, Cena Ilość</b> <b>0.08}    {Produkt,</b> <b>Ilość, Cena} &lt;-</b> <b>Koszyk].</b>	<b>ZVAT =</b> <b>[{Produkt, Ilosc,</b> <b>Cena, Cena Ilosc</b> <b>0.08}    {Produkt,</b> <b>Ilosc, Cena} &lt;-</b> <b>Koszyk].</b>
225	9	<b>Koszyk =</b> <b>[{ołówek, 4,</b> <b>0.25}, {długopis,</b> <b>1, 1.20}, {papier,</b> <b>2, 0.20}].</b>	<b>Koszyk =</b> <b>[{olowek, 4,</b> <b>0.25}, {dlugopis,</b> <b>1, 1.20}, {papier,</b> <b>2, 0.20}].</b>

229	3	<pre> - module(przetlumacz). -export([petla/0]).  petla() -&gt; receive "house" -&gt; io:format("dom~n" ), petla();  "white" -&gt; io:format("biały~n" ), petla();  _ -&gt; io:format("Nie rozumiem.~n" ), petla()  end. </pre>	<pre> - module(przetlumacz). -export([petla/0]).  petla() -&gt; receive "house" -&gt; io:format("dom~n" ), petla();  "white" -&gt; io:format("biały~n" ), petla();  _ -&gt; io:format("Nie rozumiem.~n" ), petla()  end. </pre>
233	20	<pre> Z ! "biały", </pre>	<pre> Z ! "biały", </pre>
233	22	<pre> {Z, _} -&gt; Z ! "Nie rozumiem." , loop() </pre>	<pre> {Z, _} -&gt; Z ! "Nie rozumiem." , petla() </pre>
234	29	<pre> petla() -&gt; receive 3 -&gt; io:format("bum.~n" ), exit({ruletk, die, at, erlang:time()}); _ -&gt; io:format("kliknij~n" ), loop() end. </pre>	<pre> petla() -&gt; receive 3 -&gt; io:format("bum.~n" ), exit({ruletk, die, at, erlang:time()}); _ -&gt; io:format("kliknij~n" ), petla() end. </pre>

236	11, 30	<pre> {EXIT, Od, Przyczyna} -&gt; io:format("Strzelec ~p zginął z powodu ~p." , [Od, Przyczyna]), io:format("Uruchom kolejny proces.~n" ), petla() end.</pre>	<pre> {EXIT, Od, Przyczyna} -&gt; io:format("Strzelec ~p zginął z powodu ~p." , [Od, Przyczyna]), io:format("Uruchom kolejny proces.~n" ), petla() end.</pre>
238	14	<pre> io:format("Strzelec ~p zginął z powodu ~p." , [Od, Przyczyna]), io:format(" Restart. ~n" ),</pre>	<pre> io:format("Strzelec ~p zginął z powodu ~p." , [Od, Przyczyna]), io:format(" Restart. ~n" ),</pre>
249	28	<pre> (/ 2,0 4)</pre>	<pre> (/ 2.0 4)</pre>
274	34	<pre> (iterate para- fibonacci [1 1])</pre>	<pre> (iterate para- fibonacci [1 1]))</pre>
283	21	<pre> user=&gt; (deref</pre>	<pre> user=&gt; (deref film)</pre>
285	1	<pre> user=&gt; (reset! zagrozenie "Podział bez ryzyka."))</pre>	<pre> user=&gt; (reset! zagrozenie "Podział bez ryzyka."))</pre>