

Tajniki Java 9

Pisanie reaktywnego, modularnego, współbieżnego
i bezpiecznego kodu

Dr Edward Lavieri

Peter Verhas

Przekład: Jakub Niedźwiedź

APN Promise
Warszawa 2018

Spis treści

Wprowadzenie	1
Rozdział 1: Krajobraz języka Java 9	7
Java 9 z lotu ptaka	8
Burzenie monolitu	9
Wykorzystanie powłoki Java Shell	10
Sterowanie procesami zewnętrznymi	11
Podnoszenie wydajności dzięki G1	11
Mierzenie wydajności przy pomocy JMH	11
Wprowadzenie do HTTP 2.0	12
Zastosowanie programowania reaktywnego	12
Poszerzanie listy życzeń	13
Podsumowanie	14
Rozdział 2: Odkrywanie Java 9	15
Poprawione sporne blokowanie [JEP 143]	16
Cele poprawy	16
Dzielona pamięć podręczna kodu [JEP 197]	17
Alokacja pamięci	18
Kompilacja Smart Java, faza druga [JEP 199]	18
Obsługa ostrzeżeń z narzędzi Lint i Doclint [JEP 212]	19
Warstwowe przypisywanie typów w javac [JEP 215]	20
Obsługa adnotacji 2.0 [JEP 217]	21
Nowy schemat łańcucha wersji [JEP 223]	22
Automatyczne generowanie testów kompilatora w czasie wykonywania programu [JEP 233]	23
Testowanie atrybutów plików klas generowanych przez Javac [JEP 235]	24
Przechowywanie łańcuchów tekstowych w archiwach CDS [JEP 250]	25
Problem	25
Rozwiązanie	26

Przygotowywanie kontrolek interfejsu użytkownika JavaFX oraz interfejsów CSS API do modularyzacji [JEP 253]	26
Przegląd JavaFX	27
Implikacje dla wersji Java 9	28
Zwarte łańcuchy tekstowe [JEP 254]	29
Stan sprzed wersji Java 9	30
Nowość w wersji Java 9	30
Włączanie wybranych aktualizacji Xerces 2.11.0 do JAXP [JEP 255]	30
Zaktualizowanie JavaFX/Media do nowszej wersji GStreamer [JEP 257]	31
Silnik HarfBuzz do obsługi czcionek [JEP 258]	32
Grafika HiDPI w systemach Windows i Linux [JEP 263]	33
Renderer graficzny Marlin [JEP 265]	34
Unicode 8.0.0 [JEP 267]	34
Nowości w Unicode 8.0.0	34
Zaktualizowane klasy w wersji Java 9	35
Zarezerwowane obszary stosu dla sekcji krytycznych [JEP 270]	35
Sytuacja sprzed wersji Java 9	35
Nowości w wersji Java 9	36
Dynamiczne łączenie modeli obiektowych zdefiniowanych przez język [JEP 276] .	37
Dowód poprawności	37
Dodatkowe testy dla olbrzymich obiektów w G1 [JEP 278]	38
Poprawienie rozwiązywania problemów z nieudanymi testami [JEP 279]	39
Informacje środowiskowe	40
Informacje o procesach Java	40
Optymalizowanie łączenia łańcuchów tekstowych [JEP 280]	41
Platforma HotSpot do testów jednostkowych w C++ [JEP 281]	41
Umożliwienie korzystania z GTK 3 w systemach Linux [JEP 283]	42
Nowy system budowania HotSpot [JEP 284]	43
Podsumowanie	44
Rozdział 3: Usprawnienia języka Java 9	45
Praca z dojściami do zmiennych [JEP 193]	46
Praca z narzędziami AtoMiC Toolkit	47
Użycie klasy sun.misc.Unsafe	49
Pomijanie ostrzeżeń o przestarzałych funkcjach w instrukcjach importu [JEP 211] 50	
Rozszerzanie funkcji Project Coin [JEP 213]	51

Korzystanie z adnotacji @SafeVarargs	51
Instrukcja try-with-resource	52
Korzystanie z operatora rombowego	53
Zaprzestanie użycia znaku podkreślenia	54
Użycie prywatnych metod interfejsowych	55
Poprawne przetwarzanie instrukcji import [JEP-216]	58
Podsumowanie	59
Rozdział 4: Budowanie modularnych aplikacji w Java 9	61
Wprowadzenie do modularności	62
Przegląd systemu modułów platformy Java [JEP-200]	64
Profil kompaktowy 1	64
Profil kompaktowy 2	65
Profil kompaktowy 3	65
Modularyzacja kodu źródłowego JDK [JEP-201]	68
Organizacja kodu źródłowego JDK sprzed wersji Java 9	69
Narzędzia programistyczne	70
Wdrażanie	70
Dostosowanie do warunków międzynarodowych	70
Monitorowanie	70
RMI	71
Bezpieczeństwo	71
Rozwiązywanie problemów	71
Usługi WWW	72
Narzędzia JavaFX	72
Środowisko uruchomieniowe Java	72
Kod źródłowy	72
Biblioteki	73
Pliki nagłówkowe C	74
Baza danych	75
Zreorganizowany kod źródłowy JDK	75
Zrozumienie modularnych obrazów uruchomieniowych [JEP-220]	75
Przyjęcie formatu plików uruchomieniowych	76
Zmiana struktury obrazu uruchomieniowego	76
Obsługa typowych operacji	78
Usunięcie przywilejów z klas JDK	78

Zachowywanie istniejących zachowań	78
Poznanie systemu modułów [JEP-261]	78
Ścieżki dla modułów	79
Naruszenia granic kontroli dostępu	79
Środowisko uruchomieniowe	80
Modularne pakowanie aplikacji Java [JEP-275]	82
Zaawansowane możliwości narzędzia Java Linker	82
Opcje narzędzia Java Packager	82
JLink – konsolidator Java Linker [JEP-282]	85
Hermetyzacja większości wewnętrznych interfejsów API [JEP-260]	87
Podsumowanie	88
Rozdział 5: Migrowanie aplikacji do Java 9	89
Szybki przegląd Project Jigsaw	90
Classpath	90
Monolityczna natura JDK	91
Jak moduły wpasowują się do ogólnych założeń platformy Java	92
Moduł bazowy	93
Niezawodna konfiguracja	95
Silna hermetyzacja	96
Planowanie migracji	97
Testowanie prostej aplikacji Java	97
Potencjalne problemy z migracją aplikacji	100
JRE	100
Dostęp do wewnętrznych interfejsów API	100
Dostęp do wewnętrznych pakietów JAR	101
Dezaktualizacja adresów URL dla pakietów JAR	102
Mechanizm rozszerzeń	103
Modularyzacja pakietu JDK	104
Porady od firmy Oracle	106
Kroki przygotowawcze	106
Pozyskanie zbudowanego pakietu JDK 9	106
Uruchomienie programu przed rekompilacją	106
Aktualizacja bibliotek i narzędzi firm trzecich	107
Kompilacja aplikacji	107
Opcje -source i -target przed wersją Java 9	109

Opcje -source i -target w wersji Java 9	110
Uruchomienie narzędzia jdeps dla danego kodu	110
Przerywanie hermetyzacji	113
Opcja --add-opens	114
Opcja --add-exports	114
Opcja --permit-illegal-access	114
Zmiany w obrazie uruchomieniowym	115
Schemat wersji Java	115
Układ JDK i JRE	116
Co zostało usunięte	118
Zaktualizowane odświeżanie pamięci	119
Wdrażanie	120
Wybór wersji JRE	120
Serializowane aplety	120
Aktualizacja JNLP	120
Zagnieżdżone zasoby	121
Rozszerzenie FX XML	121
Składnia pliku JNLP	123
Liczbowe porównywanie wersji	123
Przydatne narzędzia	124
Środowisko Java – jEnv	124
Maven	126
Pozyskiwanie M2Eclipse IDE	127
Podsumowanie	129
Rozdział 6: Eksperymentowanie z powłoką Java Shell	131
Czym jest JShell?	132
Początki pracy z JShell	133
Praktyczne zastosowania JShell	138
Tryby sprzężenia zwrotnego	138
Tworzenie niestandardowego trybu sprzężenia zwrotnego	143
Wypisywanie listy aktywów	145
Edytowanie w JShell	146
Modyfikowanie tekstu	146
Podstawowa nawigacja	147
Nawigacja historyczna	147

Zaawansowane polecenia edycyjne	148
Praca ze skryptami	148
Skrypty startowe	148
Ładowanie skryptów	148
Zapisywanie skryptów	149
Zaawansowane skrypty w JShell	149
Podsumowanie	151
Rozdział 7: Wykorzystanie nowego, domyślnego odśmiecania G1	153
Przegląd odśmiecania pamięci	154
Cykl życia obiektu	154
Tworzenie obiektu	154
Dalsze życie obiektu	155
Niszczenie obiektu	155
Algorytmy odśmiecania pamięci	156
Mark and sweep (oznacz i zamieć)	156
Odśmiecanie pamięci CMS	156
Szeregowe odśmiecanie pamięci	157
Równoległe odśmiecanie pamięci	157
Odśmiecanie pamięci G1	157
Opcje odśmiecania pamięci	158
Metody języka Java związane z odśmiecaniem pamięci	162
Metoda System.gc()	162
Metoda finalize()	164
Odśmiecanie pamięci przed wersją Java 9	166
Wizualizowanie odśmiecania pamięci	166
Aktualizacje odśmiecania pamięci w wersji Java 8	167
Studium przypadku – gry pisane w języku Java	168
Odśmiecanie pamięci na nowej platformie Java	169
Domyślne odśmiecanie pamięci	169
Przestarzałe kombinacje odśmiecania pamięci	171
Zunifikowane rejestrowanie działań odśmiecania pamięci	172
Zunifikowane rejestrowanie JVM (JEP-158)	173
Znaczniki	173
Poziomy	174
Dekoracje	174

Informacje wyjściowe.....	175
Opcje wiersza poleceń.....	175
Zunifikowane rejestrowanie odśmiecania pamięci (JEP-271)	175
Opcje rejestrowania odśmiecania pamięci	176
Znacznik gc	178
Makra.....	178
Dodatkowe uwarunkowania	179
Utrzymujące się problemy	179
Sprawianie, aby obiekty podlegały odśmiecaniu pamięci	180
Podsumowanie	182
Rozdział 8: Mikroanalizowanie aplikacji przy pomocy JMH	183
Przegląd mikroanalizowania	184
Podejście do korzystania z JMH	185
Instalowanie Java 9 i Eclipse ze wsparciem dla Java 9	185
Praktyczny eksperyment	187
Mikroanalizowanie przy pomocy Maven.....	189
Opcje analizowania	194
Tryby.....	194
Jednostki czasu	195
Techniki pomagające uniknąć pułapek mikroanalizowania.....	195
Zarządzanie energią.....	196
Systemowe programy planujące	196
Współdzielenie czasu.....	196
Eliminowanie ślepych uliczek w kodzie i zawijania stałych	196
Rozbieżność pomiędzy uruchomieniami.....	197
Pojemność pamięci podręcznej	198
Podsumowanie	198
Rozdział 9: Wykorzystanie interfejsu API ProcessHandle.....	199
Czym są procesy?	200
Nowy interfejs ProcessHandle	201
Uzyskiwanie identyfikatora PID bieżącego procesu	202
Uzyskiwanie informacji na temat procesu	202
Wypisywanie listy procesów.....	204
Wypisywanie listy procesów podrzędnych.....	204

Wypisywanie listy procesów potomnych	205
Wypisywanie listy wszystkich procesów	206
Oczekiwanie na procesy	207
Kończenie procesów	208
Niewielka aplikacja sterująca procesami	211
Klasa Main	211
Klasa Parameters	213
Klasa ParamsAndHandle	214
Klasa ControlDaemon	214
Podsumowanie	218
Rozdział 10: Dokładne śledzenie stosu	219
Przegląd stosu Java	219
Znaczenie informacji o stosie	220
Przykład – ograniczanie elementów wywołujących	222
Przykład – uzyskiwanie obiektu rejestrującego dla elementu wywołującego	225
Praca z klasą StackWalker	226
Pozyskiwanie wystąpienia klasy StackWalker	226
RETAIN_CLASS_REFERENCE	227
SHOW_REFLECT_FRAMES	227
SHOW_HIDDEN_FRAMES	227
Końcowe uwagi na temat stałych wyliczenia	230
Dostęp do klas	230
Metody do przechodzenia przez stos	231
StackFrame	233
Wydajność	234
Podsumowanie	234
Rozdział 11: Nowe narzędzia i usprawnienia narzędzi	235
Nowy klient HTTP [JEP-110]	236
Klient HTTP przed wersją Java 9	236
Nowy klient HTTP w wersji Java 9	239
Ograniczenia nowego interfejsu API	240
Uproszczony interfejs API Doclet [JEP-221]	242
Interfejs API Doclet przed wersją Java 9	242
Wyliczenia interfejsu API	244

Klasy interfejsu API	244
Interfejsy API	245
Problemy z istniejącym wcześniej interfejsem API Doclet	246
Interfejs API Doclet w wersji Java 9	246
Interfejs API drzewa kompilatora	246
Interfejs API modelu języka	248
Interfejs AnnotatedConstruct	248
Wyliczenie SourceVersion	253
Wyjątek UnknownEntityException	254
Javadoc z HTML5 [JEP-224]	255
Wyszukiwanie w Javadoc [JEP-225]	260
Wprowadzenie wyszukiwania z użyciem pierwszych liter słów	261
Usunięcie wyboru wersji JRE [JEP-231]	261
Interfejs API Parser dla silnika Nashorn [JEP-236]	262
Nashorn	262
Korzystanie z Nashorn jako narzędzia wiersza poleceń	263
Korzystanie z Nashorn jako osadzonego interpretera	266
ECMAScript	267
Interfejs API Parser	267
Pliki JAR dla wielu wersji [JEP-238]	268
Identyfikowanie plików JAR dla wielu wersji	270
Powiązane zmiany w pakiecie JDK	271
Interfejs kompilatora JVM na poziomie Java [JEP-243]	272
Adnotacje BeanInfo [JEP-256]	273
JavaBean	273
BeanProperty	274
SwingContainer	274
Klasy BeanInfo	275
Wejście/wyjście dla obrazów TIFF [JEP-262]	275
Interfejs API i usługa rejestrowania platformowego [JEP-264]	278
Pakiet java.util.logging	278
Rejestrowanie w wersji Java 9	280
Katalogi XML [JEP-268]	281
Standard OASIS XML Catalog	281
Procesory JAXP	282
Katalogi XML przed wersją Java 9	282

Zmiany na platformie Java 9	282
Wygodne metody fabryczne dla kolekcji [JEP-269]	283
Wykorzystanie kolekcji przed wersją Java 9	284
Wykorzystanie nowych literałów dla kolekcji	286
Funkcje biurkowe specyficzne dla platformy [JEP-272]	287
Ulepszona obsługa metod [JEP-274]	288
Powody dla wprowadzenia poprawek	288
Funkcje wyszukiwawcze	288
Obsługa argumentów	289
Dodatkowe kombinacje	289
Ulepszone wycofywanie przestarzałych elementów [JEP-277]	290
Co tak naprawdę oznacza adnotacja @Deprecated	291
Podsumowanie	291
Rozdział 12: Współbieżność i programowanie reaktywne	293
Programowanie reaktywne	294
Standaryzacja programowania reaktywnego	295
Nowy interfejs API Flow	297
Interfejs Flow.Publisher	297
Interfejs Flow.Subscriber	298
Interfejs Flow.Subscription	298
Interfejs Flow.Processor	298
Przykładowa implementacja	299
Dodatkowe aktualizacje współbieżności	300
Współbieżność na platformie Java	300
Wyjaśnienie pojęcia współbieżności	301
Konfiguracje systemowe	301
Wątki na platformie Java	302
Ulepszenia współbieżności	305
Usprawnienia interfejsu API CompletableFuture	307
Szczegóły klasy	307
Ulepszenia	311
Wskazówki dla cykli spin-wait	312
Podsumowanie	312

Rozdział 13: Usprawnienia zabezpieczeń	313
Protokół Datagram Transport Layer Security	314
Protokół DTLS wersja 1.0	314
Protokół DTLS wersja 1.2	316
Wsparcie dla DTLS w wersji Java 9	319
Tworzenie magazynów kluczy PKCS12	320
Wprowadzenie do magazynów kluczy	320
Java Keystore (JKS)	320
Builder	321
Klasa CallbackHandlerProtection	322
Klasa PasswordProtection	322
Klasa PrivateKeyEntry	323
Klasa SecretKeyEntry	323
Klasa TrustedCertificateEntry	323
Domyślne ustawienie PKCS12 w Java 9	325
Poprawienie wydajności zabezpieczeń aplikacji	325
Wymuszanie zasad zabezpieczeń	325
Obliczanie uprawnień	326
Pakiet java.Security.CodeSource	327
Algorytm sprawdzania pakietów	327
Rozszerzenie negocjacyjne dla protokołu TLS w warstwie aplikacji	328
Rozszerzenie ALPN dla TLS	328
Pakiet javax.net.ssl	329
Rozszerzenie pakietu java.net.ssl	330
Wykorzystanie instrukcji procesora w algorytmach GHASH i RSA	332
Funkcje skrótu	332
Podpinanie OCSP dla TLS	334
Podstawy podpinania OCSP	334
Zmiany dla platformy Java 9	335
Implementacje SecureRandom oparte na DRBG	337
Podsumowanie	337
Rozdział 14: Flagi wiersza poleceń	339
Zunifikowane rejestrowanie JVM [JEP 158]	340
Opcje wiersza poleceń	341

Dekoracje	343
Poziomy	344
Informacje wyjściowe	344
Znaczniki	344
Sterowanie kompilatorem [JEP 165]	345
Tryby kompilacji	345
Tryb kompilacji C1	346
Tryb kompilacji C2	346
Kompilacja warstwowa	346
Sterowanie kompilatorem w wersji Java 9	347
Polecenia diagnostyczne [JEP 228]	349
Agent profilowania sterty [JEP 240]	350
Usunięcie narzędzia JHAT [JEP 241]	351
Sprawdzanie poprawności argumentów flag wiersza poleceń JVM [JEP 245]	352
Kompilacja dla starszych wersji platformy [JEP 247]	353
Podsumowanie	354
Rozdział 15: Najlepsze praktyki w Java 9	357
Wsparcie dla UTF-8	357
Klasa ResourceBundle	358
Zagnieżdżona klasa	359
Pola i konstruktory	363
Metody	363
Zmiany w wersji Java 9	369
Unicode 7.0.0	369
Pakiet java.lang	370
Pakiet java.text	370
Dodatkowe znaczenie	371
Przeniesienie pakietu JDK na platformę Linux/AArch64	372
Obrazy z wieloma rozdzielczościami	373
Repozytorium Common Locale Data Repository (CLDR)	374
Podsumowanie	375
Rozdział 16: Przyszłe kierunki rozwoju	377
Przyszłe zmiany w pakiecie JDK	378
Zmiany w JDK docelowe dla wersji Java 10	378

Konsolidacja repozytoriów	378
Usunięcie narzędzia do generowania rodzimych nagłówków	379
Zgłoszone propozycje związane z JDK	380
Zrównoleglenie fazy pełnego odśmiecania pamięci w CMS	380
Interfejsy REST API dla JMX	381
Wsparcie dla alokacji sterty	382
Szkicowe propozycje związane z JDK	382
Przyspieszenie finalizowania obiektów	382
Model pamięci Java	383
Interfejsy obcych funkcji	384
Metody izolowane	384
Ograniczenie marnowania metaprzestrzeni	384
Poprawienie obsługi IPv6	385
Rozpakowane listy argumentów dla dojść do metod	386
Ulepszone demo MandelbrotSet, wykorzystujące typy wartościowe	387
Wydajne działanie porównywania tablic	388
Przyszłe zmiany w kompilatorze Java	388
Zasady wygaszania opcji -source i -target dla javac	388
Statyczne analizatory z obsługą wtyczek	388
Przyszłe zmiany w Java Virtual Machine	389
Zgłoszone propozycje związane z JVM	389
Platforma Java działająca na kontenerach	389
Umożliwienie wykonywania metod języka Java na procesorach GPU	390
Epsilon GC – Niskokosztowe odśmiecanie pamięci	391
Szkicowe propozycje związane z JVM	391
Zapewnienie stabilnych punktów próbnikowych USDT w metodach skompiłowanych dla JVM	392
Zawężanie współbieżnego monitora	392
Zapewnienie niskokosztowego sposobu próbkowania alokacji na sterce Java	393
Platforma poleceń diagnostycznych	394
Ulepszone redefiniowanie klas	394
Domyślne włączanie trybu NUMA, gdy jest to wskazane	394
Obiekty typów wartościowych	395
Zasady sprawdzania dostępu w JVM	396
Przyszłe zmiany w JavaX	396
Adnotacje specyficzne dla JMX do rejestrowania zarządzanych zasobów	396

Modernizacja implementacji wyglądu i zachowania GTK3	397
Trwające projekty specjalne	397
Annotations pipeline 2.0	399
Audio Synthesis Engine	399
Caciocavallo	399
Common VM Interface	399
Compiler Grammar	399
Da Vinci Machine	400
Device I/O	400
Graal	400
HarfBuzz Integration	400
Kona	401
OpenJFX	401
Panama	401
Shenandoah	401
Podsumowanie	402