

Spis treści

Wstęp

Podziękowania

O autorze

1. Podstawowe struktury programistyczne

1.1. Nasz pierwszy program

1.1.1. Analiza programu Witaj, świecie!

1.1.2. Kompilacja i uruchamianie programu w języku Java

1.1.3. Wywołania metod

1.1.4. JShell

1.2. Typy proste

1.2.1. Typy całkowite ze znakiem

1.2.2. Typy zmiennoprzecinkowe

1.2.3. Typ char

1.2.4. Typ boolean

1.3. Zmienne

1.3.1. Deklaracje zmiennych

1.3.2. Nazwy

1.3.3. Inicjalizacja

1.3.4. Stałe

1.4. Działania arytmetyczne

1.4.1. Przypisanie

1.4.2. Podstawowa arytmetyka

1.4.3. Metody matematyczne

1.4.4. Konwersja typów liczbowych

1.4.5. Operatory relacji i operatory logiczne

1.4.6. Duże liczby

1.5. Ciągi znaków

1.5.1. Łączenie ciągów znaków

1.5.2. Wycinanie ciągów znaków

1.5.3. Porównywanie ciągów znaków

1.5.4. Konwersja liczb na znaki i znaków na liczby

1.5.5. API klasy String

1.5.6. Kodowanie znaków w języku Java

1.6. Wejście i wyjście

1.6.1. Wczytywanie danych wejściowych

1.6.2. Formatowanie generowanych danych

1.7. Kontrola przepływu

1.7.1. Instrukcje warunkowe

1.7.2. Pętle

1.7.3. Przerywanie i kontynuacja

1.7.4. Zasięg zmiennych lokalnych

1.8. Tablice i listy tablic

1.8.1. Obsługa tablic

1.8.2. Tworzenie tablicy

1.8.3. Klasa ArrayList

1.8.4. Klasy opakowujące typy proste

1.8.5. Rozszerzona pętla for

1.8.6. Kopiowanie tablic i obiektów ArrayList

1.8.7. Algorytmy tablic

1.8.8. Parametry wiersza poleceń

1.8.9. Tablice wielowymiarowe

1.9. Dekompozycja funkcjonalna

1.9.1. Deklarowanie i wywoływanie metod statycznych

1.9.2. Parametry tablicowe i zwracane wartości

1.9.3. Zmienna liczba parametrów

Ćwiczenia

2. Programowanie obiektowe

2.1. Praca z obiektami

2.1.1. Metody dostępne i modyfikujące

2.1.2. Referencje do obiektu

2.2. Implementowanie klas

2.2.1. Zmienne instancji

2.2.2. Nagłówki metod

2.2.3. Treści metod

2.2.4. Wywołania metod instancji

2.2.5. Referencja this

2.2.6. Wywołanie przez wartość

2.3. Tworzenie obiektów

2.3.1. Implementacja konstruktorów

2.3.2. Przeciążanie

2.3.3. Wywoływanie jednego konstruktora z innego

2.3.4. Domyślna inicjalizacja

2.3.5. Inicjalizacja zmiennych instancji

2.3.6. Zmienne instancji z modyfikatorem final

2.3.7. Konstruktor bez parametrów

2.4. Statyczne zmienne i metody

2.4.1. Zmienne statyczne

2.4.2. Stałe statyczne

2.4.3. Statyczne bloki inicjalizacyjne

2.4.4. Metody statyczne

2.4.5. Metody wytwórcze

2.5. Pakiety

2.5.1. Deklarowanie pakietów

2.5.2. Polecenie jar

2.5.3. Ścieżka przeszukiwań dla klas

2.5.4. Dostęp do pakietu

2.5.5. Importowanie klas

2.5.6. Import metod statycznych

2.6. Klasy zagnieżdżone

2.6.1. Statyczne klasy zagnieżdżone

2.6.2. Klasy wewnętrzne

2.6.3. Specjalne reguły składni dla klas wewnętrznych

2.7. Komentarze do dokumentacji

2.7.1. Wstawianie komentarzy

2.7.2. Komentarze klasy

2.7.3. Komentarze metod

2.7.4. Komentarze zmiennych

2.7.5. Ogólne komentarze

2.7.6. Odnośniki

2.7.7. Opisy pakietów, modułów i ogólne

2.7.8. Wycinanie komentarzy

Ćwiczenia

3. Interfejsy i wyrażenia lambda

3.1. Interfejsy

3.1.1. Deklarowanie interfejsu

3.1.2. Implementowanie interfejsu

- 3.1.3. Konwersja do typu interfejsu
- 3.1.4. Rzutowanie i operator instanceof
- 3.1.5. Rozszerzanie interfejsów
- 3.1.6. Implementacja wielu interfejsów
- 3.1.7. Stałe
- 3.2. Metody statyczne, domyślne i prywatne
 - 3.2.1. Metody statyczne
 - 3.2.2. Metody domyślne
 - 3.2.3. Rozstrzyganie konfliktów metod domyślnych
 - 3.2.4. Metody prywatne
- 3.3. Przykłady interfejsów
 - 3.3.1. Interfejs Comparable
 - 3.3.2. Interfejs Comparator
 - 3.3.3. Interfejs Runnable
 - 3.3.4. Wywołania zwrotne interfejsu użytkownika
- 3.4. Wyrażenia lambda
 - 3.4.1. Składnia wyrażeń lambda
 - 3.4.2. Interfejsy funkcyjne
- 3.5. Referencje do metod i konstruktora
 - 3.5.1. Referencje do metod
 - 3.5.2. Referencje konstruktora
- 3.6. Przetwarzanie wyrażeń lambda
 - 3.6.1. Implementacja odroczonego wykonania
 - 3.6.2. Wybór interfejsu funkcjonalnego
 - 3.6.3. Implementowanie własnych interfejsów funkcjonalnych
- 3.7. Wyrażenia lambda i zasięg zmiennych
 - 3.7.1. Zasięg zmiennej lambda

3.7.2. Dostęp do zmiennych zewnętrznych

3.8. Funkcje wyższych rzędów

3.8.1. Metody zwracające funkcje

3.8.2. Metody modyfikujące funkcje

3.8.3. Metody interfejsu Comparator

3.9. Klasy lokalne i anonimowe

3.9.1. Klasy lokalne

3.9.2. Klasy anonimowe

Ćwiczenia

4. Dziedziczenie i mechanizm refleksji

4.1. Rozszerzanie klas

4.1.1. Klasy nadrzędne i podrzędne

4.1.2. Definiowanie i dziedziczenie metod klas podrzędnych

4.1.3. Przesłanianie metod

4.1.4. Tworzenie klasy podrzędnej

4.1.5. Przypisania klas nadrzędnych

4.1.6. Rzutowanie

4.1.7. Metody i klasy z modyfikatorem final

4.1.8. Abstrakcyjne metody i klasy

4.1.9. Ograniczony dostęp

4.1.10. Anonimowe klasy podrzędne

4.1.11. Dziedziczenie i metody domyślne

4.1.12. Wywołania metod z super

4.2. Object najwyższa klasa nadrzędna

4.2.1. Metoda toString

4.2.2. Metoda equals

4.2.3. Metoda hashCode

4.2.4. Klonowanie obiektów

4.3. Wyliczenia

4.3.1. Sposoby wyliczania

4.3.2. Konstruktory, metody i pola

4.3.3. Zawartość elementów

4.3.4. Elementy statyczne

4.3.5. Wyrażenia switch ze stałymi wyliczeniowymi

4.4. Informacje o typie i zasobach w czasie działania programu

4.4.1. Klasa Class

4.4.2. Wczytywanie zasobów

4.4.3. Programy wczytujące klasy

4.4.4. Kontekstowy program wczytujący klasy

4.4.5. Programy do ładowania usług

4.5. Refleksje

4.5.1. Wyliczanie elementów klasy

4.5.2. Kontrolowanie obiektów

4.5.3. Wywoływanie metod

4.5.4. Tworzenie obiektów

4.5.5. JavaBeans

4.5.6. Praca z tablicami

4.5.7. Klasa Proxy

Ćwiczenia

5. Wyjątki, asercje i logi

5.1. Obsługa wyjątków

5.1.1. Wyrzucanie wyjątków

5.1.2. Hierarchia wyjątków

5.1.3. Deklarowanie wyjątków kontrolowanych

5.1.4. Przechwytywanie wyjątków

5.1.5. Wyrażenie try z określeniem zasobów

5.1.6. Klauzula finally

5.1.7. Ponowne wyrzucanie wyjątków i łączenie ich w łańcuchy

5.1.8. Nieprzechwycone wyjątki i ślad stosu wywołań

5.1.9. Metoda Objects.requireNonNull

5.2. Asercje

5.2.1. Użycie asercji

5.2.2. Włączanie i wyłączanie asercji

5.3. Rejestrowanie danych

5.3.1. Klasa Logger

5.3.2. Mechanizmy rejestrujące dane

5.3.3. Poziomy rejestrowania danych

5.3.4. Inne metody rejestrowania danych

5.3.5. Konfiguracja mechanizmów rejestrowania danych

5.3.6. Programy obsługujące rejestrowanie danych

5.3.7. Filtry i formaty

Ćwiczenia

6. Programowanie uogólnione

6.1. Klasy uogólnione

6.2. Metody uogólnione

6.3. Ograniczenia typów

6.4. Zmienność typów i symbole wieloznaczne

6.4.1. Symbole wieloznaczne w typach podrzędnych

6.4.2. Symbole wieloznaczne typów nadrzędnych

6.4.3. Symbole wieloznaczne ze zmiennymi typami

6.4.4. Nieograniczone symbole wieloznaczne

6.4.5. Przechwytywanie symboli wieloznacznych

6.5. Uogólnienia w maszynie wirtualnej Javy

6.5.1. Wymazywanie typów

6.5.2. Wprowadzanie rzutowania

6.5.3. Metody pomostowe

6.6. Ograniczenia uogólnień

6.6.1. Brak typów prostych

6.6.2. W czasie działania kodu wszystkie typy są surowe

6.6.3. Nie możesz tworzyć instancji zmiennych opisujących typy

6.6.4. Nie możesz tworzyć tablic z parametryzowanym typem

6.6.5. Zmienne opisujące typ klasy nie są poprawne w kontekście statycznym

6.6.6. Metody nie mogą wywoływać konfliktów po wymazywaniu typów

6.6.7. Wyjątki i uogólnienia

6.7. Refleksje i uogólnienia

6.7.1. Klasa `Class<T>`

6.7.2. Informacje o uogólnionych typach w maszynie wirtualnej

Ćwiczenia

7. Kolekcje

7.1. Mechanizmy do zarządzania kolekcjami

7.2. Iteratory

7.3. Zestawy

7.4. Mapy

7.5. Inne kolekcje

7.5.1. Właściwości

7.5.2. Zestawy bitów

7.5.3. Zestawy wyliczeniowe i mapy

7.5.4. Stosy, kolejki zwykłe i dwukierunkowe oraz kolejki z priorytetami

7.5.5. Klasa WeakHashMap

7.6. Widoki

7.6.1. Małe kolekcje

7.6.2. Zakresy

7.6.3. Niemodyfikowalne widoki

Ćwiczenia

8. Strumienie

8.1. Od iteratorów do operacji strumieniowych

8.2. Tworzenie strumienia

8.3. Metody filter, map i flatMap

8.4. Wycinanie podstrumieni i łączenie strumieni

8.5. Inne przekształcenia strumieni

8.6. Proste redukcje

8.7. Typ Optional

8.7.1. Jak korzystać z wartości Optional

8.7.2. Jak nie korzystać z wartości Optional

8.7.3. Tworzenie wartości Optional

8.7.4. Łączenie flatMap z funkcjami wartości Optional

8.7.5. Zamiana Optional w Stream

8.8. Kolekcje wyników

8.9. Tworzenie map

8.10. Grupowanie i partycjonowanie

8.11. Kolektory strumieniowe

8.12. Operacje redukcji

8.13. Strumienie typów prostych

8.14. Strumienie równoległe

Ćwiczenia

9. Przetwarzanie danych wejściowych i wyjściowych

9.1. Strumienie wejściowe i wyjściowe, mechanizmy wczytujące i zapisujące

9.1.1. Pozyskiwanie strumieni

9.1.2. Wczytywanie bajtów

9.1.3. Zapisywanie bajtów

9.1.4. Kodowanie znaków

9.1.5. Wczytywanie danych tekstowych

9.1.6. Generowanie danych tekstowych

9.1.7. Wczytywanie i zapisywanie danych binarnych

9.1.8. Pliki o swobodnym dostępie

9.1.9. Pliki mapowane w pamięci

9.1.10. Blokowanie plików

9.2. Ścieżki, pliki i katalogi

9.2.1. Ścieżki

9.2.2. Tworzenie plików i katalogów

9.2.3. Kopiowanie, przenoszenie i usuwanie plików

9.2.4. Odwiedzanie katalogów

9.2.5. System plików ZIP

9.3. Połączenia HTTP

9.3.1. Klasy URLConnection i HttpURLConnection

9.3.2. API klienta HTTP

9.4. Wyrażenia regularne

9.4.1. Składnia wyrażeń regularnych

9.4.2. Odnajdywanie pojedynczego dopasowania

9.4.3. Odnajdywanie wszystkich dopasowań

9.4.4. Grupy

9.4.5. Dzielenie za pomocą znaczników

9.4.6. Zastępowanie dopasowań

9.4.7. Flagi

9.5. Serializacja

9.5.1. Interfejs Serializable

9.5.2. Chwilowe zmienne instancji

9.5.3. Metody readObject i writeObject

9.5.4. Metody readResolve i writeReplace

9.5.5. Wersjonowanie

Ćwiczenia

10. Programowanie współbieżne

10.1. Zadania współbieżne

10.1.1. Uruchamianie zadań

10.1.2. Obiekty Future

10.2. Obliczenia asynchroniczne

10.2.1. Klasa CompletableFuture

10.2.2. Tworzenie obiektów typu CompletableFuture

10.2.3. Długie zadania obsługujące interfejs użytkownika

10.3. Bezpieczeństwo wątków

10.3.1. Widoczność

10.3.2. Wyścigi

10.3.3. Strategie bezpiecznego korzystania ze współbieżności

10.3.4. Klasy niemodyfikowalne

10.4. Algorytmy równoległe

10.4.1. Strumienie równoległe

10.4.2. Równoległe operacje na tablicach

10.5. Struktury danych bezpieczne dla wątków

10.5.1. Klasa ConcurrentHashMap

10.5.2. Kolejki blokujące

10.5.3. Inne struktury danych bezpieczne dla wątków

10.6. Atomowe liczniki i akumulatory

10.7. Blokady i warunki

10.7.1. Blokady

10.7.2. Słowo kluczowe synchronized

10.7.3. Oczekiwanie warunkowe

10.8. Wątki

10.8.1. Uruchamianie wątku

10.8.2. Przerywanie wątków

10.8.3. Zmienne lokalne w wątku

10.8.4. Dodatkowe właściwości wątku

10.9. Procesy

10.9.1. Tworzenie procesu

10.9.2. Uruchamianie procesu

10.9.3. Uchwyty procesów

Ćwiczenia

11. Adnotacje

11.1. Używanie adnotacji

11.1.1. Elementy adnotacji

11.1.2. Wielokrotne i powtarzane adnotacje

11.1.3. Adnotacje deklaracji

11.1.4. Adnotacje wykorzystania typów

11.1.5. Jawne określanie odbiorców

11.2. Definiowanie adnotacji

11.3. Adnotacje standardowe

11.3.1. Adnotacje do kompilacji

11.3.2. Adnotacje do zarządzania zasobami

11.3.3. Metaadnotacje

11.4. Przetwarzanie adnotacji w kodzie

11.5. Przetwarzanie adnotacji w kodzie źródłowym

11.5.1. Przetwarzanie adnotacji

11.5.2. API modelu języka

11.5.3. Wykorzystanie adnotacji do generowania kodu źródłowego

Ćwiczenia

12. API daty i czasu

12.1. Linia czasu

12.2. Daty lokalne

12.3. Modyfikatory daty

12.4. Czas lokalny

12.5. Czas strefowy

12.6. Formatowanie i przetwarzanie

12.7. Współpraca z przestarzałym kodem

Ćwiczenia

13. Internacjonalizacja

13.1. Lokalizacje

13.1.1. Określanie lokalizacji

13.1.2. Domyślna lokalizacja

13.1.3. Nazwy wyświetlane

13.2. Formaty liczb

13.3. Waluty

13.4. Formatowanie czasu i daty

13.5. Porównywanie i normalizacja

13.6. Formatowanie komunikatów

13.7. Pakiety z zasobami

13.7.1. Organizacja pakietów z zasobami

13.7.2. Klasy z pakietami

13.8. Kodowanie znaków

13.9. Preferencje

Ćwiczenia

14. Kompilacja i skryptowanie

14.1. API kompilatora

14.1.1. Wywołanie kompilatora

14.1.2. Uruchamianie zadania kompilacji

14.1.3. Wczytywanie plików źródłowych z pamięci

14.1.4. Zapisywanie skompilowanego kodu w pamięci

14.1.5. Przechwytywanie komunikatów diagnostycznych

14.2. API skryptów

14.2.1. Tworzenie silnika skryptowego

14.2.2. Powiązania

14.2.3. Przekierowanie wejścia i wyjścia

14.2.4. Wywoływanie funkcji i metod skryptowych

14.2.5. Kompilowanie skryptu

14.3. Silnik skryptowy Nashorn

14.3.1. Uruchamianie Nashorna z wiersza poleceń

14.3.2. Wywoływanie metod pobierających i ustawiających dane oraz metod przeładowanych

14.3.3. Tworzenie obiektów języka Java

14.3.4. Ciągi znaków w językach JavaScript i Java

14.3.5. Liczby

14.3.6. Praca z tablicami

14.3.7. Listy i mapy

14.3.8. Wyrażenia lambda

14.3.9. Rozszerzanie klas Java i implementowanie interfejsów Java

14.3.10. Wyjątki

14.4. Skrypty powłoki z silnikiem Nashorn

14.4.1. Wykonywanie poleceń powłoki

14.4.2. Uzupełnianie ciągów znaków

14.4.3. Wprowadzanie danych do skryptu

Ćwiczenia

15. System modułów na platformie Java

15.1. Koncepcja modułu

15.2. Nazywanie modułów

15.3. Modularny program Witaj, świecie!

15.4. Dołączanie modułów

15.5. Eksportowanie pakietów

15.6. Moduły i dostęp przez refleksje

15.7. Modularne pliki JAR

15.8. Moduły automatyczne i moduł unnamed

15.9. Flagi wiersza poleceń dla migracji

15.10. Wymagania przechodnie i statyczne

15.11. Wybiórcze eksportowanie i otwieranie

15.12. Wczytywanie usługi

15.13. Narzędzia do pracy z modułami

Ćwiczenia