

Wprowadzenie, historia i podstawy tworzenia pierwszego programu w Javie

JAVA to rodzaj języka programowania, który został zaprojektowany specjalnie do użycia w Internecie. Jest łatwiejszy w użyciu niż jego odpowiednik, C ++, a jednocześnie wykorzystuje model programowania zorientowanego obiektowo. JAVA może być wykorzystywana do tworzenia całych aplikacji, które mogą być rozprowadzane wśród klientów i serwerów w sieci lub działać na pojedynczym komputerze. JAVA również może być służyć do budowania małego apletu lub małego modułu aplikacji, który ma być używany jako część strony internetowej. Aplety umożliwiają internautom interakcję z tą konkretną stroną internetową. Programy JAVA można znaleźć na urządzeniach mobilnych, komputerach stacjonarnych, dyskach Blu-ray, serwerach i inteligentnych kartach. JAVA pierwotnie był znany jako Oak, kiedy został po raz pierwszy stworzony. Oak został stworzony przez Jamesa Goslinga w 1991 roku. James Gosling pracował dla firmy w tym czasie znanej jako Sun Microsystems, a pierwotnym celem języka programowania Goslinga miał być stosowany w małych urządzeniach elektronicznych, takich jak telewizory, tostery, magnetowidy i tym podobne. naprzód. Celem JAVA od samego początku miała być szybka, wydajna, mała i przenośna. Sun Microsystems zmienił nazwę na język programowania i wprowadził JAVA w 1995 roku. Po tym wprowadzeniu możliwości interaktywne sieci World Wide Web wzrosły wykładniczo. Maszyna wirtualna JAVA jest zawarta we wszystkich głównych przeglądarkach internetowych. Podobnie niemal każdy znaczący programista systemu operacyjnego dodał kompilatory JAVA jako część tego, co oferują w swoich produktach, a głównymi programistami systemów operacyjnych są takie firmy, jak Microsoft i IBM. JAVA jest obecnie własnością i jest zarządzana przez firmę o nazwie Oracle. Jak więc powstało imię JAVA? Nazwa JAVA powstała, gdy James Gosling wraz ze swoimi członkami zespołu opracowywał język JAVA i pili obficie duże ilości kawy. Zespół JAVA uważał, że ze względu na dużą ilość i wysoką jakość kawy, którą piją, byli w stanie lepiej poprawić jakość języka programowania. Kawa, którą James Gosling i jego zespół pili, miała szczególną rolę w rozwoju języka programowania JAVA i została rozprowadzona na cały świat z miejsca o nazwie "Java Island". Zasadniczo Gosling i jego najnowocześniejszy zespół zostali zainspirowani do napisania nowego języka programowania nazwą miejsca, w którym powstał ich kofeinowy nektar. Stąd symbolem języka programowania JAVA jest niewątpliwy kubek i spodek

Najważniejsze 10 faktów, które musisz wiedzieć o Javie

1. Termin Java odnosi się ogólnie do kombinacji trójkowej tych rzeczy:

- Język programowania znany jako JAVA

-JVM lub wirtualna maszyna JAVA, która jest platformą JAVA. JVM jest tym, który uruchamia skompilowane kody bajtowe w Javie, zwykle na zestawie standardowych bibliotek specyficznych dla JAVA. Biblioteki te obejmują te dostarczane przez edycję Enterprise, określaną jako EE lub Java Standard Edition, która jest określana jako SE. Chociaż platforma JAVA i JVM są zaprojektowane tak, aby współpracować ze sobą, sam język nie musi oznaczać użycia wirtualnej maszyny JAVA i na odwrót.

-Wirtualna maszyna Java (JVM) odnosi się do maszyny wirtualnej działającej na bardzo wysokim poziomie wydajności i wykonuje kody bajtów na bardzo specyficznej platformie

komputerowej. Maszyna wirtualna JAVA jest skrócona i jest po prostu określana przez programistów zazwyczaj jako JVM.

2. Większość wyższych warstw powszechnie stosowanego systemu operacyjnego Android lub systemu operacyjnego jest zbudowana przy użyciu JAVA, dlatego konieczne jest poznanie JAVA, jeśli Twoim celem jest ostatecznie tworzenie aplikacji dla systemu operacyjnego Android. Dzięki programowaniu JAVA możesz tworzyć gry, systemy wbudowane, strony internetowe i aplikacje komputerowe, które mogłyby być podobne do wersji tych programów na urządzeniach mobilnych i innych urządzeniach.

3. Dostępne są 4 różne platformy JAVA:

- The JAVA Micro Edition. Wersja JAVA Micro Edition zawiera biblioteki i frameworki potrzebne do tworzenia aplikacji działających na mniejszych urządzeniach, zwanych urządzeniami mikro, takimi jak tablety i smartfony.

- Wersja JAVA Enterprise Edition. Wersja JAVA Enterprise Edition składa się z bibliotek i frameworków, które są wymagane do tworzenia aplikacji, które są wykorzystywane głównie przez przedsiębiorstwa.

- Wersja JAVA FX Edition. JAVA FX Edition składa się z bibliotek graficznych, które są używane do pracy na różnych platformach, a także do tworzenia silnych aplikacji klienckich.

- Wersja standardowa JAVA. Wersja Standard Edition JAVA składa się ze wszystkich podstawowych funkcji JAVA i wszystkich podstawowych bibliotek JAVA. Jest to wydanie, którego będziesz używał do nauki programowania JAVA tutaj, a także z wykorzystaniem czasu lub kontynuacji nauki programowania.

4. Język programowania JAVA jest zorientowanym obiektowo językiem programowania, określanym przez programistów jako OOP i działa na wysokim poziomie.

5. Programy JAVA są uruchamiane na specjalnym dedykowanym oprogramowaniu znanym jako wirtualna maszyna JAVA, określanym jako JVM w świecie programowania.

6. Maszyna wirtualna JAVA lub JVM jest obsługiwana przez dużą liczbę systemów operacyjnych. Z uwagi na to, że JAVA może działać na tak wielu systemach operacyjnych, czyni to JAVA językiem wieloplatformowym. To stąd pochodzi popularne zdanie "napisz raz, uciekaj w dowolnym miejscu" lub WORA. Termin ten jest również czasem nazywany "napisz raz, uruchom wszędzie" lub WORE.

7. Aby zainstalować maszynę wirtualną JAVA i uruchomić programy JAVA, należy pobrać, a następnie zainstalować środowisko wykonawcze JAVA, znane również jako środowisko JRE. Zauważ, że środowiska JRE nie można używać do tworzenia programów JAVA. JRE można używać tylko do odczytu programów JAVA.

8. Aby uzyskać dostęp do narzędzi niezbędnych do rozwoju aplikacji JAVA, należy najpierw pobrać i zainstalować zestaw rozwojowy JAVA. Zestaw programistyczny JAVA jest również określanym przez programistów jako JDK. Najnowsza dostępna wersja JDK jest wersją

8. Nie musisz jednak pobierać środowiska JRE, jeśli pobierzesz wersję JDK 8, ponieważ możesz swobodnie rozpowszechniać środowisko JRE w tej wersji ze swoją aplikacją, zgodnie z warunkami licencji JRE.

9. Akronim IDE odnosi się do pakietu oprogramowania, który znany jest jako Zintegrowane Środowisko Rozwoju i zapewnia ludziom, którzy programują komputery, niezbędne kompleksowe narzędzia potrzebne do tworzenia oprogramowania, testowania i ogólnego rozwoju. Zwykle nowoczesne IDE składa się z debuggera, edytora kodu źródłowego oraz interpretera lub kompilatora. Deweloper oprogramowania korzysta z jednego graficznego interfejsu użytkownika, o którym prosi deweloperów po prostu GUI, w celu uzyskania dostępu do tych trzech narzędzi. Istnieją dwa bardzo popularne IDE JAVA, które są całkowicie darmowe - IDE znany jako Netbeans i IDE znany jako Eclipse. MY pokażemy, jak programować przy użyciu IDEclipse IDE.

10. Doskonały zasób wiedzy na temat początku programowania JAVA jest tuż przed Tobą :)

Funkcje JAVA

Główne cechy JAVA to:

* Kod JAVA jest solidny. Oznacza to, że obiekty JAVA mogą zawierać zerowe odniesienia do innych znanych obiektów lub dane zewnętrzne. Natomiast programy napisane w C ++ i innych językach nie mają tej możliwości. Proces ten zapewnia, że instrukcja nie zawiera adresu przechowywanych danych w samym systemie operacyjnym lub w innej aplikacji. Każde z tych zdarzeń spowodowałoby awarię systemu operacyjnego lub programu, lub zakończyć. Aby zapewnić integralność, wirtualna maszyna JAVA lub JVM ma wiele kontroli.

* Aplety JAVA mają inne funkcje, które działają w taki sposób, aby działały jak najskuteczniej, a nie zostały wykonane serwer, ale zamiast tego, u klient.

* JAVA jest dużo łatwiejsza do nauczenia niż C ++, chociaż język z pewnością nie jest do nauczenia z dnia na dzień.

* Programy tworzone w języku JAVA są przenośne w sieci. Oznacza to, że programy te mogą być używane w innych systemach operacyjnych niż ten, który je utworzył, bez konieczności wykonywania skomplikowanych, czasochłonnych, dużych przeróbek. Program źródłowy jest kompilowany do tego, co JAVA określa jako kod bajtowy. Ten kod bajtowy może być uruchamiany w dowolnej sieci na kliencie lub serwerze z JVM. Ta JVM następnie tłumaczy kod bajtowy do kodu, który następnie będzie mógł być uruchamiany na samym sprzęcie komputerowym, co oznacza, że wyraźne różnice w platformach, na przykład długość instrukcji, mogą być rozpoznawane i rozwiązywane lokalnie - tak szybko w miarę wykonywania programu. Zasadniczo JAVA zapewnia, że nie jest już konieczne tworzenie specyficznych dla platformy wersji oprogramowania.

* Programowanie JAVA jest określane jako "zorientowane obiektowo". Oznacza to, że pojedynczy obiekt może być częścią grupy innych obiektów, a także dziedziczyć kod wspólny dla tej klasy i wiele innych funkcji. Zamiast tradycyjnych proceduralnych "czasowników", przedmioty są określane w terminach "rzeczowników", z którymi jest bardziej

prawdopodobne, że ktoś jest bardziej związany. Metoda może być postrzegana jako jedno z zachowań lub możliwości obiektu.

Więcej informacji o programowaniu obiektowym (OOP)

Programowanie obiektowe, określane przez programistów jako OOP, jest modelem języka zbudowanym wokół danych zamiast logiki, a także obiektów zamiast działań. Po przetworzeniu danych wejściowych przekazuje dane wyjściowe. OOP uważa, że obiekt zainteresowania może być przetwarzany zgodnie z tą logiką jest potrzebny, aby działały. Obiekty te, na przykład, mogą różnić się od widżetów komputerowych do budynku i jego podłóg dla ludzi. W OOP, pierwszym krokiem jest zlokalizowanie wszystkich obiektów, którymi programista chce manipulować, a następnie określenie, w jaki sposób te konkretne obiekty mogą się ze sobą nawzajem odnosić. Ten proces jest zwykle nazywany modelowaniem danych. Po wybraniu obiektu, który ma zostać zidentyfikowany, programista uogólnia go. Na przykład ta jedna konkretna książka będzie oznaczać wszystkie książki. To uogólnienie określa typ informacji przechowywanych przez obiekt, a także sekwencje logiczne, które mogą z nim pracować. Każda oddzielna sekwencja nazywana jest "metodą". Obiekty w OOP komunikują się z komunikatami, które są wyraźnie zdefiniowanymi interfejsami. Zasady, parametry i koncepcje wykorzystywane w OOP oferują te korzyści, które wszyscy programiści muszą znać i uchwycić:

- Koncepcja klasy danych umożliwia zdefiniowanie podklas obiektów, które mają wszystkie lub co najmniej niektóre cechy grupy kluczy. Ta szczególna jakość OOP, zwana dziedziczeniem, skraca czas programowania. Dziedziczenie zapewnia dokładniejsze kodowanie, a dziedziczenie wymaga również dokładniejszej analizy danych.
- Klasa definiuje tylko określone dane. Gdy ten obiekt lub grupa jest uruchomiona, kod nie będzie mylnie uzyskiwać dostępu do danych z programu, który nie jest przeznaczony. Ta szczególna cecha znana jako "ukrywanie danych" pozwala uniknąć niezamierzonego uszkodzenia danych i zapewnia większe bezpieczeństwo systemu.
- Skoro definicja pewnej klasy jest nie tylko zdolna do ponownego wykorzystania przez program, do którego jest przeznaczona, ale także przez inne programy zorientowane obiektowo, wówczas jest bardziej wydajnie dystrybuowana do użytku sieciowego. (Odwołaj się do zasady "pisz raz, użyj nigdzie")
- Koncepcja klasy danych umożliwia programiście tworzenie dowolnych nowych danych, które nie zostały jeszcze zdefiniowane w ramach parametrów samego języka.

Poznaj aplety

JAVA produkuje programy uruchamiane w przeglądarce, które nazywane są apletami. Mimo że aplety stają się coraz bardziej przestarzałe w miarę upływu czasu, są wykorzystywane do ułatwiania komunikacji między użytkownikami sieci i graficznym interfejsem użytkownika,

lub GUI. Zanim dostępne były aplety JAVA, strony i witryny sieci Web zwykle nie były interaktywne i poza tym nie były statyczne. Takie konkurencyjne produkty, takie jak Microsoft Silverlight i Adobe Flash, sprawiły, że JAVA stała się mniej popularna, ponieważ kiedyś

zdominowała rynek. JAVA uruchamia aplety w przeglądarce internetowej za pomocą JVM. JVM następnie tłumaczy dany kod na tak zwane "natywne instrukcje procesora", znane również jako kod natywny. To tłumaczenie umożliwia pośredni program platformy lub pośrednie wykonanie systemu operacyjnego (OS). JAVA Virtual Machine dostarcza większość komponentów niezbędnych do uruchomienia kodu bajtowego. Ten kod bajtowy jest zwykle znacznie mniejszy niż inne języki programowania, w których programy wykonywalne są zwykle zapisywane. Zasadniczo, gdy jest używany online, JAVA pozwala na używanie apletów, a także pobieranie ich przez przeglądarkę. Proces ten umożliwia przeglądarce dostęp do funkcji lub wykonywanie funkcji, która normalnie nie byłaby dostępna bez apletu. Program lub aplet musi zostać pobrany lub zainstalowane przez użytkownika, zanim użytkownik będzie w stanie uzyskać pełny dostęp i korzystać z programu JAVA

Czym różni się JAVA od JavaScript

Chociaż zarówno JAVA, jak i JavaScript są językami programowania używanymi do tworzenia funkcji lub aplikacji na stronie internetowej, bardzo ważne jest, aby JAVA nie mylił się z JavaScriptem. JavaScript pochodzi z Netscape i jest nieco łatwiejszy do nauczenia niż JAVA. JavaScript jest również interpretowany na wyższym poziomie niż JAVA, co oznacza, że JavaScript wykorzystuje język programowania z silną abstrakcją lub techniką zarządzania złożonością systemów komputerowych, od szczegółów komputera. Jednak JavaScriptowi brakuje prędkości kodu bajtowego JAVA, a także pewnej przenośności. Oznacza to, że szybkość działania JavaScriptu jest znacznie wolniejsza niż w przypadku JAVA. Aplety JAVA mogą działać niemal na każdym systemie operacyjnym bez konieczności ponownej kompilacji i nie mają żadnych odmian ani rozszerzeń, które są unikalne dla systemu operacyjnego. To właśnie sprawia, że JAVA jest ogólnie uważany za najbardziej strategiczny język w tworzeniu aplikacji internetowych. JavaScript może być jednak przydatny w bardzo małych aplikacjach działających na serwerze WWW lub w kliencie WWW. Najczęściej zauważalną różnicą między JAVA i JavaScriptem jest typ aplikacji, w których są one używane. Programy Java są wykorzystywane do aplikacji zainicjowanych przez stronę internetową lub uruchamianych z pulpitu komputera. Programy te są samodzielnymi programami, które zwykle otwierają całkowicie oddzielne okno programu. Ponadto komputer absolutnie nie jest w stanie uruchomić aplikacji JAVA bez zainstalowanego JAVA. JavaScript jest zawsze zawarty we wszystkich aktualnych przeglądarkach i uruchamia JavaScript na stronie internetowej podczas wczytywania strony internetowej, o ile włączona jest obsługa JavaScript. Kiedy programista używa kodu JavaScript, "myślenie" JavaScript i obliczenia są konsekwentnie zawsze wykonywane po stronie klienta. Strona klienta odnosi się do komputera, na którym początkowo uzyskiwana jest strona w sieci. Z drugiej strony, programy JAVA normalnie wykonują wszystkie główne "myślenie" i proces obliczeń w ramach określonego apletu JAVA, który najpierw musi zostać pobrany, lub po stronie serwera. Z tego powodu JavaScript działa po stronie klienta, Zwykle działa szybciej niż JAVA, a czasami nawet prawie natychmiast, w zależności od szybkości połączenia. Ponieważ programy JAVA są uruchamiane po stronie klienta, przetwarzanie zajmuje trochę więcej czasu, ale zwykle trwa tylko kilka sekund lub dłużej. JavaScript używa znacznie mniej pamięci niż JAVA, nawet bardzo, bardzo mało pamięci w niektórych przypadkach, aby wykonać jego przetwarzanie, a także działać poprawnie. Ze względu na mniejsze wymagania dotyczące pamięci i możliwość oferowania tak wielu różnych

funkcji stron internetowych, JavaScript jest bardzo popularnym językiem programowym używanym na wielu różnych stronach w Internecie w tym czasie. Programy JAVA mogą jednak czasami wymagać użycia dużej ilości pamięci komputera w celu prawidłowego działania. To wymaganie może bezwzględnie spowodować, że inny program będzie działał znacznie wolniej, a nawet spowolni działanie komputera. Tak więc nawet podczas gdy programy JAVA mają wrodzoną zdolność do rozwijania się do robienia różnych, często potężnych rzeczy, ponieważ wykorzystuje większe wykorzystanie pamięci, może to być również niekorzystna prędkość pod pewnymi względami.

Jak JAVA jest dziś używana

Od 2016 roku JAVA jest obecnie wykorzystywana przez ponad 9 milionów programistów na całym świecie i jest również jednym z najczęściej używanych języków programowania, szczególnie dla aplikacji klient-serwer. Bez względu na architekturę komputera, aplikacje JAVA można skompilować w kod, który działa na wszystkich maszynach JVM. JAVA była identyfikowana głównie z niską wydajnością, z koniecznością oczekiwania na uaktualnienia Sun Microsystems, z interpretacją kodu bajtowego oraz z użyciem Apletów. Dzisiaj jednak JAVA stała się bardziej związana z niezawodnie wysoką wydajnością, z dynamiczną kompilacją hotspotów, z coraz bardziej niezależną społecznością open source, z architekturą zorientowaną na usługi, SOA, z usługami sieciowymi i z aplikacjami internetowymi. W niniejszym rozdziale omówiono zastosowania JVM, języka programowania JAVA, a także platform JAVA.

Używanie języka programowania JAVA

Język programowania JAVA działa na wysokim poziomie i był inspirowany przez program Smalltalk oraz programowanie w języku C i C ++. Język programowania JAVA również wypożyczył kilka pomysłów z innych języków. Syntaktyczny projekt języka programowania JAVA był znany tym, którzy już znają się na "nawiasach klamrowych", językach zstępujących z C, ale mają silniejsze (niekiedy bezspornie) zasady OO niż zasady używane w C ++. Ponadto, zbieranie śmieci jest dość automatyczne, co eliminuje potrzebę, aby programista zwolnił pamięć używaną przez obiekty, które stały się przestarzałe i nie są już używane. JAVA jest filozoficznie określany jako język "fail early". Ponieważ ma ograniczenia składni, większość błędów programistycznych po prostu nie jest w ogóle możliwa z Javą. Z uwagi na fakt, że JAVA nie ma bezpośredniego dostępu do wskaźników, błędy specyficzne dla jego arytmetyki po prostu nie istnieją. Wykorzystanie pewnego obiektu jako typu innego niż ten, który pierwotnie nazwano, wymaga prostej konwersji odlewu, co pozwala twórcy kodu na oczywistą szansę na odmowę programowania, które może być nielogiczne, a tym samym niemożliwe do zrealizowania. Zróżnicowana liczba frameworków Java dla przedsiębiorstw powoduje konieczność używania deskryptorów wdrażania lub plików konfiguracyjnych, które zazwyczaj są zapisywane w formacie XML w celu zidentyfikowania funkcji. Może to obejmować kolejność kroków wymaganych do wykonania w regule lub która klasa może obsłużyć określone żądanie w HTTP. Innymi słowy, muszą one wykraczać poza język, aby właściwie realizować swoją funkcję. JAVA 5.0 dodaje przypisy do tego języka, co pozwala na oznaczanie klas, pól i metod wartościami, które następnie mogą być rozwijane i kontrolowane w czasie ich uruchamiania, a najczęściej za pomocą refleksji. Wielu programistów preferuje te przypisy, zwykle określone

jako adnotacje, ponieważ są one w stanie uprościć rzeczy, które mogą być wywoływane przez niektóre deskryptory lub inne sposoby. Są one również w stanie utrudnić interpretację kodu Java. Mimo to przypis lub przypis może mieć wpływ na wykonanie kodu, a także może wpływać na to w sposób, który może nie być całkowicie oczywisty z przypisu lub sama adnotacja. Poza tą jedyną krytyką, JAVA jest uważana głównie za najpopularniejszy język programowania ogólnego zastosowania, wykorzystywany obecnie w świecie. Zalety języka programowania JAVA są niezwykle rozległe. Główne korzyści obejmują, ale z pewnością nie są ograniczone do:

* Korzyści z ogromnie ogromnej bazy wiedzy.

* Korzyści z ogromnej ilości programistów, którzy są zawsze łatwo dostępni.

* Korzyści z bezpłatnych narzędzi, które są powszechnie dostępne na wielu platformach, takich jak Windows, Solaris, Linux i Mac. Wszystkie te platformy są w stanie wykonywać i kompilować wszystkie aplikacje Java.

Sam język programowania JAVA osiąga unikalny i wartościowy punkt w quid pro quo między wydajnością kodu a wydajnością programisty. Podczas gdy cykle CPU (jednostka centralna) ciągle zmniejszają współczynnik kosztu, twórcy oprogramowania zwykle nie kontynuują zmniejszyć koszty. Jeśli tak, koszt twórców oprogramowania rośnie z wielu powodów, między innymi rosnąca inflacja i wzrost technicznych kosztów kształcenia. Z tego powodu może to nieuchronnie doprowadzić do akceptacji jeszcze jednego fragmentu abstrakcji pomiędzy opcodes CPU realizacja i sami programiści oprogramowania - o ile pozwala to programistom na tworzenie oprogramowania o wyższej jakości i znacznie bardziej wydajnym tempie.

Wyjaśnienie korzystania z platform JAVA

JAVA jest postrzegana ogólnie w kategoriach trzech różnych platform. Te trzy platformy zostały pokrótce omówione wcześniej w tej książce. Zwróć uwagę, że JAVA FX Edition nie jest tutaj uwzględniona.

1. Wersja Enterprise, znana również jako JAVA EE

2. Wersja Standard Edition, znana również jako JAVA SE

i

3. The Micro Edition, znany również jako JAVA ME

Każda z tych różnych platform wykorzystuje kombinację określonej grupy bibliotek, maszyny wirtualnej i wersji językowej, która służy do wykonywania kodu. Wersja EE zawiera wszystko, co robi wersja SE i więcej, co oznacza, że każda aplikacja EE może uzyskać dostęp do całej biblioteki SE i korzystać z niej. Również użycie EE w języku programowania jest identyczne jak w SE. Oznacza to, że wersja SE jest podzbiorem wersji EE. JAVA Micro Edition lub ME różni się znacznie od swoich odpowiedników ze względu na to, że małe urządzenie (takie jak smartfon) jest po prostu znacznie bardziej ograniczone pod względem miejsca i możliwości. Micro Edition nie może być uważany za podzbiór Standard Edition (dlatego nie może być uważany za podzbiór EE), ponieważ wiele jego bibliotek istnieje tylko w Micro Edition. Ponadto wersja Micro Edition całkowicie eliminuje niektóre funkcje języka programowania z powodu

ograniczeń obliczeniowych platformy, na której jest uruchamiana. Przykłady niektórych wyeliminowanych funkcji obejmują klasę Float i prymitywę float. To oznacza, że Micro Edition wymaga nieco innego zestawu narzędzi niż wersja Enterprise Edition i wersja Standard Edition. Przy tak głębokich przesunięciach w urządzeniach, całkowita przenośność kodu jest znacznie mniej realistyczna w mikroprzestrzeni. Faktycznie, wielu programistów Javy postrzega ME jako zupełnie inne zwierzę, prawie osobną jednostkę nawet.

Czego można się spodziewać podczas korzystania z wirtualnej maszyny JAVA (JVM)

Kod źródłowy JAVA musi w pewnym momencie zostać przetłumaczony na natywny kod wykonywalny platformy. Zwykle wymaga to zastosowania procesu dwuetapowego:

1. Twórca oprogramowania musi skompilować kod źródłowy do kodu JAVA.
2. JVM musi następnie przetłumaczyć te informacje na temat używanej platformy na natywny kod tej platformy.

Drugi krok był pierwotnie wykonywany przez interpretację lub przez wzięcie każdej pojedynczej instrukcji JAVA Virtual Machine (JVM) i konwersję tej instrukcji w jednej chwili do jednej lub więcej instrukcji natywnych dla platformy. Następnie, po wykonaniu tego, kompilatory w czasie, o których wspominali twórcy oprogramowania, jako kompilatory JIT, konwertują program z kodu bajtowego na JVM na kod natywny dla tej konkretnej platformy w trakcie działania programu. Istnieje kilka sposobów osiągnąć to nawrócenie w obecnym czasie. Kompilator HotSpot firmy Sun Microsystems inicjuje ten proces, tłumacząc kod, a następnie profilując go w środowisku wykonawczym, optymalizując i kompilując poszczególne obszary, które uznano za najważniejsze, aby ten program mógł nadal działać wydajnie. IBM "interpreter trybów mieszanych" swoich maszyn JVM działa w bardzo podobny sposób. Te procedury omijają opóźnienie wydajności rozruchowej spowodowane użyciem kompilatorów JIT w całym programie, ale to także oznacza, że wydajność jest kumulowana, przybывая konsekwentnie przez pewien czas, kiedy kluczowe sekcje kodu są wykrywane i uruchamiane w bardziej efektywny sposób. Z powodu tej procedury aplikacje klienta są znacznie mniej wykorzystywane przez to podejście niż długo działające procesy serwera.

Czego można się spodziewać podczas używania JAVA bez używania wirtualnej maszyny JAVA

Całkowicie można uruchomić JAVA bez użycia wirtualnej maszyny JAVA. Ponieważ źródło JAVA kończy się przekształcaniem w kod bajtowy, to z kolei staje się rodzimym kodem platformy, tak jak to się dzieje, w rzeczywistości można to osiągnąć jednocześnie. GNU (GNU rekursywnie odnosi się do terminu "GNU's Not Unix"). Kompilator dla JAVA, określany również jako GCJ, umożliwia z góry, jednorazowe kompendium kodu JAVA do polecenia, które może być wykonane na pojedynczej platformie. Chociaż istnieje wystarczająca ilość informacji do kompilowania aplikacji działających w linii poleceń i po stronie serwera podczas korzystania z tego procesu, w rzeczywistości nie jest to możliwe pozwala na obsługę zestawu abstrakcyjnych okien dialogowych (AWT), co oczywiście czyniłoby go nieprzydatnym do programowania AWT, a także do programowania GUI Swing. Ta oczywista wada korzystania z tego procesu jest to, że kod wieloplatformowy staje się związany tylko z jedną platformą i tylko w jednym kroku. Co więcej, ten statyczny typ kompilacji nie neguje automatycznie działania dynamicznej

kompilacji HotSpota. Autor wspomnianej kompilacji wykonał pracę nad badaniem, w którym wynik GCJ został uznany za mniej niż zaledwie 5% powyżej wzrostu wydajności wersji HotSpot. Mimo to GCJ ma przewagę pod względem wydajności i ma zdolność rozwiązywania kluczowych problemów, takich jak możliwość zwięzłego wdrażania konkretnej aplikacji Java, która może być w pełni uruchomiona niezależnie od wersji JVM i czy jest nawet dostępne.

Czego można się spodziewać podczas korzystania z wirtualnej maszyny JAVA bez JAVA

Jest również absolutnie możliwe, aby od razu przejść do poziomu JVM i pomyślnie całkowicie ominąć język JAVA. Kompilatory kodu bajtowego C-do-JVM już istnieją; kompilatory takie jak Axiomatic Multi-Platform C, które jest komercyjne i dostarcza częściową wersję ANSI C. Co więcej, postęp manipulacji kodem bajtowym JAVA pozwala na tworzenie klas w czasie wykonywania, które są całkowicie wykonywane w aplikacjach Java. Oznacza to, że to, z czym będziesz pracować, nie będzie już JAVA, ale będzie efektywną formą języka programowania używanego do montażu w wirtualnej maszynie JAVA.

Wyjaśnienie procesu społecznościowego JAVA

Istnieje społeczność JAVA, która istnieje poza maszynami wirtualnymi, poza bibliotekami i poza językiem programowania. Pomimo przytłaczającej ilości oprogramowania, które jest uważane za open source i wykorzystywane przez i napisane w języku programowania JAVA, istnieje ciągle oczywisty i otwarty konflikt między społecznością open source a społecznością JAVA. Proces wspólnotowy JAVA, nazywany czasem przez programistów jako po prostu JCP, został ustanowiony w 1998 r. JCP jest sformalizowanym mechanizmem umożliwiającym zainteresowanym stronom wniesienie wkładu w opracowywanie standardowych specyfikacji technicznych technologii JAVA. Jest otwarty dla publiczności i każdy może zostać członkiem procesu społecznościowego JAVA, po prostu przesyłając odpowiedni formularz dostępny na swojej stronie internetowej. Bardzo zróżnicowana liczba aspektów JAVA istnieje całkowicie bez standaryzowanych akceptacji JCP. Wiele niezależnych projektów zyskało rozmiary wystarczające do faktycznego konkurowania z rzeczywistymi benchmarkami JCP na poziomie świadomości w świadomości konsumentów JCP. Jako całość jednak te niezależne projekty mogły również szybko przystosować się do modyfikacji całkowicie bez użycia JAVA, a jednocześnie wykorzystywać najlepsze cechy tych modyfikacji, podobnie jak w funkcji AJAX, która jest funkcją upraszczającą kod. Jest to projekt Direct Web Remoting, DWR lub inspirowany Railsami Trails.

Wprowadzenie do Podstawy programowania JAVA

Jeśli chodzi o naukę języka programowania JAVA lub jakiegokolwiek innego języka programowania dla tej sprawy, istnieje pięć podstawowych pojęć, które musisz zrozumieć, zanim zaczniesz. Te pięć podstawowych pojęć obejmuje:

1. Zmienne
2. Struktury danych
3. Struktury kontrolne
4. Składnia

5. Narzędzia

Każda z tych pięciu koncepcji zostanie dokładnie wyjaśniona w tej książce na poziomie początkującym, aby zapewnić ich zrozumienie.

Zmienne

Zmienne są w rzeczywistości kamieniem węgielnym wszelkich programów JAVA, w które napotkasz, i dlatego są podstawą języka programowania JAVA. Z definicji zmienną w programowaniu komputerowym jest po prostu lokalizacja pamięci oraz związana z nią nazwa, która symbolizuje nieznaną lub znaną informację lub ilość. Innymi słowy, konkretna wartość. Mówiąc najprościej, zmienna jest metodą przechowywania pewnego rodzaju informacji do wykorzystania w późniejszym czasie i jest dostępna do odzyskania przez odniesienie do nazwy lub słowa, które będą opisywać te informacje. Załóżmy na przykład, że wybierasz nawigację do określonej witryny, a pierwszą rzeczą, którą robi ta witryna, jest zapytanie o to, jak się nazywasz. Zwykle robi się to, aby wprowadzić jakąś formę ludzkiej znajomości - podczas twojej następnej wizyty ta strona zadzwoni do ciebie po imieniu. Osoba, która zbudowała witrynę, utworzy na ekranie małe pole tekstowe, które będzie wymagało podania imienia i nazwiska, a małe pole tekstowe będzie oznaczać zmienną. Osoba, która zbudowała stronę, może zdecydować się nazwać to małe pole tekstowe czymś w stylu "Visitor Name", a to będzie symboliczne słowo lub nazwa małej zmiennej w polu tekstowym. Tak więc po wpisaniu swojego imienia i nazwiska w małym polu tekstowym, twoje imię i nazwisko jest następnie przechowywane jako informacja w zmiennej o nazwie "Visitor Name". Ta informacja byłaby łatwo dostępna dla osoby, która zbudowała stronę lub programisty. Wtedy ta osoba będzie mogła ponownie odwiedzić stronę i zapytać "Jaką wartość ma zmienna" Nazwa odwiedzającego "?", A program odpowie programistom wartością tego, co wpisałeś w tym małym polu tekstowym, zobaczyłem, kiedy po raz pierwszy odwiedziłeś stronę. Ta koncepcja jest ciągle używana przez cały czas programowania JAVA i jest również bardzo skuteczna w programowaniu. Ta właśnie koncepcja sprawia, że Twitter i Facebook działają; to właśnie pozwala ci zapłacić swoje rachunki za pomocą aplikacji bankowości internetowej lub bankowej, a także umożliwia licytowanie w serwisach takich jak eBay. Zmienne pozwalają światu programistycznemu obracać się w kółko. Okay, teraz nadszedł czas na trochę bardziej szczegółowe. Istnieją różne rodzaje zmiennych, jeśli chodzi o programowanie przy użyciu języka programowania JAVA. Jeśli programista zdecyduje się zapisać twoją nazwę w zmiennej, twoja nazwa będzie przechowywana jako typ zmiennej o nazwie String. A może programista chciał również przechowywać twój wiek - który byłby przechowywany jako typ zmiennej o nazwie Integer. W końcu, być może programista chciał przechowywać twój roczny dochód - twój dochód byłby wtedy przechowywany jako rodzaj zmiennej zwanej Double. Podsumowując, omówiono tu trzy różne typy zmiennych:

łańcuchowy

Liczba całkowita

Double

Czym dokładnie jest zmienna String, zmienna Integer i zmienna Double? Podczas korzystania z JAVA język programowania musi wiedzieć, jakiego rodzaju informacje będą przechowywane w określonej zmiennej. Język programowania musi to wiedzieć, ponieważ JAVA jest językiem mocno napisanym. Zasadniczo, aby dany język był słabo typowanym, oznacza to, że typy wszystkich zmiennych są wnioskowane lub znane w momencie kompilacji. Jednak język słabo typowanym napisany nie pozwala na użycie jednego typu jako innego, co oznacza, że wykorzystuje różne typy zmiennych. W tym miejscu wykorzystywane są zmienne łańcuchowe, całkowite i double

Zmienne łańcuchowe

Wpisywanie w języku programowania JAVA pozwala językowi programowania na pewno wiedzieć, że informacje przechowywane w zmiennej będą konkretnie zdefiniowane. Kiedy w JAVA mowa jest o łańcuchu, patrzymy na dane tak, jakby były po prostu zdaniami składającymi się ze słów w języku angielskim. Ciąg tylko konkretnie reprezentuje litery umieszczone w określonej kolejności. Podobnie jak w języku angielskim, zmienne łańcuchowe stanowią ciąg liter ułożony w określonej kolejności, co nadaje tej serii liter bardzo konkretne znaczenie. W ten sposób, że zmienne łańcuchowe stają się bardziej zrozumiałe przez porównanie z językiem potocznym, dodanie dwóch ciągów razem działa bardzo podobnie. Jeśli masz dwie zmienne żądań i przechowywane, na przykład, dane "Visitor" i "Name", jeśli zostały dodane razem, otrzymasz ciąg: "VisitorName".

Zmienne całkowite

W języku JAVA zmienna całkowita oznacza, że masz liczbę, która nie zawiera miejsc dziesiętnych. Oznacza to liczbę całkowitą, na przykład 23 lub -784. W Javie, gdy określono, że zmienna jest liczbą całkowitą, po prostu nie można przechowywać niczego oprócz liczby całkowitej. Na przykład, jeśli chcesz dodać dwie liczby razem, liczba 35 i liczba 5. Java faktycznie zachowuje się naprawdę inaczej, zależy to od dokładnego typu zmiennej używanej do przechowywania tych danych. Następnie dodanie 35 i 5 razem spowoduje, że liczba całkowita 40. Zamiast tylko łączenia dwóch, jako ciąg (gdy 35 i 5 zostaną dodane razem w ciągu, otrzymasz 355), liczby całkowite są dodawane razem w prostej matematyce .

Zmienna double

Podwójna zmienna w JAVA będzie albo używać bardzo dużych liczb, albo bardzo małych liczb. Wartości minimalne oraz maksymalne wynoszą odpowiednio 17, a następnie 307 zer. Podwójna zmienna może również zawierać wartości punktów znane jako "podnoszące". Wartość zmiennoprzecinkowa to w zasadzie dowolna wartość liczbowa z kropką dziesiętną. Najważniejsze jest to, że posiadanie typu pomoże ci zrozumieć, jakie rzeczy możesz zrobić z informacjami zawartymi w zmiennej. Typy zmiennych są naprawdę potężne i mają sens, co jest dozwolone, a co nie jest dozwolone w obrębie pewnej grupy zmiennych

W jaki sposób wykorzystywane są struktury danych w języku programowania JAVA

Struktura danych w informatyce jest specyficzną metodą organizowania i przechowywania informacji w komputerze, dzięki czemu może być wykorzystywana w najbardziej efektywny sposób. Aby lepiej wyjaśnić struktury danych, użyjmy pojęcia listy kontaktów. Zwykle lista

kontaktów zawiera liczne kontakty, które w każdej chwili mogą się zmniejszyć lub wzrosnąć. W tym przykładzie powiedzmy, że musisz utrzymywać pięć kontaktów w kolejności. Jeśli próbujesz reprezentować te kontakty jako zmienne w programie, musisz znać strukturę danych. W tym przypadku masz listę kontaktów. Podczas programowania JAVA istnieje struktura danych nazywana listą. Oto jak wygląda kod:

1. Lista kontaktów = new ArrayList ();

Nie martw się jeszcze o symbole, które zostaną omówione później. Wszystko, co musisz wiedzieć w tej chwili, to to, że istnieje sposób na zapisanie listy w strukturze danych. W strukturze danych listy możesz łatwo dodawać elementy do listy, a także usuwać elementy z listy. `contacts.add ("John Doe (john.doe@somename.com)"); contacts.add ("Jane Doe (jane.doe@somename.com)");` Podczas tworzenia struktury danych niezwykle wydajne jest utworzenie jednej zmiennej, jak pokazano powyżej, zamiast innej zmiennej, którą należy zapisać dla każdego elementu na liście. Ponieważ stworzyłeś tylko jedną zmienną, `contacts.add (RandomContact)`, co oznacza, że twój kod jest bardziej dynamiczny i elastyczny. W tym przypadku dynamiczna odnosi się do faktu, że wynik programu może się zmieniać w zależności od wprowadzanych zmiennych. Idealnie chcesz, aby kod był renderowany tak dynamicznie, jak to tylko możliwe, a także chcesz, aby Twój kod był w stanie obsłużyć wiele różnych sytuacji bez konieczności ciągłego pisania coraz więcej kodów, w miarę upływu czasu. Zasadniczo struktura danych jest po prostu sposobem na uniknięcie konieczności tworzenia większej liczby zmiennych, niż jest to absolutnie konieczne.

W jaki sposób wykorzystywane są struktury kontrolne w języku programowania JAVA

Struktura kontrolna to blok programowania, który analizuje zmienne i wybiera przepływ w oparciu o dane parametry. Jest to podstawowy proces, w którym podejmowane są decyzje w zakresie informatyki. Kontrola przepływu reguluje sposób, w jaki komputer zareaguje, gdy zostanie mu nadane określone warunki, a także określone parametry. Oznacza to, że podczas działania określonego programu kod jest czytany przez komputer po linii. Ten proces jest znany jako "przepływ kodu". Ponieważ kod jest czytany od góry do dołu, może osiągnąć punkt, który wymaga pewnej ilości procesu decyzyjnego. Ta decyzja może spowodować, że kod przeskoczy do innej części tego konkretnego programu i może wymagać ponownego uruchomienia określonego kodu lub może po prostu pominąć wiele kodu w odpowiedzi. Ten proces można by porównać do książki "wybierz własną przygodę". Sięgasz na stronę 12 książki i masz wybór pomiędzy wyborem A i wyborem B. Program komputerowy działa w bardzo podobny sposób, z wyjątkiem tego, że program ma ścisły zestaw reguł do przestrzegania, podany przez programistę. Decyzja, że program powoduje wpływ przepływu kodu i jest znana jako struktura kontrolna.

Jak wykorzystywana jest składnia w języku programowania JAVA

Składnia języka programowania JAVA jest po prostu zestawem reguł, które definiują kombinacje symboli uważanych za precyzyjnie ustrukturyzowane programy w tym ustawionym języku. Składnia to w zasadzie szczególny układ symboli. W języku JAVA przykładem może być nawias klamrowy `{}` lub okrągły nawias `()`. Na przykład podczas przeglądania adresu e-mail można zidentyfikować fakt, że jest to adres e-mail ze względu na

składnię. Możesz także zidentyfikować to jako swoją strukturę lub po prostu "sposób, w jaki wygląda". Możesz rozpoznać adres e-mail i odróżnić go od adresu strony internetowej z powodu różnej składni, która się w nim znajduje. Składnia w języku programowania JAVA jest podobna. Istnieją zasady, a kiedy są przestrzegane, zasady te pozwalają językowi programowania zrozumieć funkcjonujące oprogramowanie, a także pozwalają tworzyć funkcjonujące oprogramowanie. Otrzymasz błędy, jeśli nie będziesz przestrzegać reguł specyficznej składni języków programowania. Podczas korzystania z JAVA istnieją cztery konkretne kroki do składni tworzenia zmiennej. Jak następuje:

1. Pierwszą częścią składni tworzenia zmiennej jest użycie słowa String, które jest typem zmiennej. Ciąg w tym konkretnym przypadku pozwala na przechowywanie zwykłych liter, jak również przechowywanie znaków specjalnych.
2. Drugą częścią składni tworzenia zmiennej jest użycie nazwy zmiennej. Nazwa zmiennej może składać się z cyfr i liter, ale jedynymi znakami specjalnymi, które są dozwolone, są podkreślenia. Pierwsza litera w nazwach zmiennych jest zwykle pisana małymi literami, co niekoniecznie jest koniecznością, ale stanowi pewną regułę w świecie programowania JAVA.
3. Trzecia część składni tworzenia zmiennej polega na wprowadzeniu wartości, którą posiada określona zmienna. W JAVA ciągi są wyraźnie zdefiniowane poprzez umieszczanie cudzysłowów wokół zwykłych liczb, liter i znaków specjalnych.
4. Czwarta i ostatnia część składni tworzenia zmiennej implementuje znak zakończenia. W języku JAVA użyto średnika. Chociaż istnieje kilka wyjątków, prawie wszystkie linie kodu JAVA zakończą się średnikiem; podobnie jak okres na końcu zdania.

Wyjaśnienie użycia narzędzi w języku programowania JAVA

Narzędzie używane w programowaniu jest bardzo podobne do każdego innego narzędzia, które my, jako ludzie, wykorzystujemy w tym, aby pomóc ci osiągnąć cel bardziej efektywnie - z tym wyjątkiem, że narzędzia JAVA są oczywiście użyteczne tylko dla programistów JAVA. W języku JAVA narzędzie to oprogramowanie, które umożliwia zwiększenie wydajności programu szybko. Istnieje szeroki wachlarz narzędzi używanych w programowaniu JAVA, ale dla wszystkich zamierzeń i celów ta książka skupi się na IDE. IDE to w zasadzie program, który w zasadzie znacznie ułatwia kodowanie i jest najbardziej istotnym narzędziem każdego programisty na świecie. IDE sprawdzają składnię kodu, aby upewnić się, że nie ma błędów. IDE również uporządkuje swoje pliki i daje im konkretny sposób na ich wszechstronne przeglądanie. IDE zwykle zawierają kodowanie, które w niektórych typowych scenariuszach wypełnia kod. IDE umożliwiają łatwą nawigację po kodzie. Istnieje wiele innych zalet korzystania z IDE, chociaż najbardziej użyteczne są tutaj wymienione, a będziesz lepiej zaznajomiony z używaniem IDE, kiedy ustawisz swój, postępując zgodnie z prostym przewodnikiem krok po kroku opisanym w części 5 tej książki.

Zapoznanie się z instrukcjami JAVA

Ogromna liczba poleceń JAVA jest zbyt liczna, aby uwzględnić każdy z nich na liście, ale ta część zawiera wizualnie niektóre z najbardziej powszechnych i użytecznych poleceń dostępnych do tej pory. Schematy w tej części są przedstawiane jako wprowadzenie do nazwy najbardziej podstawowych poleceń, a także jako odniesienie wizualne, gdy przechodzisz przez następną część, dzięki czemu będziesz mieć wiele sposobów sprawdzania swojej pracy i lepszego sensu przeczytałem do tej pory. Te polecenia JAVA nie są w żadnej szczególnej kolejności, a jedynym wyjątkiem jest pierwsze polecenie wyjaśnione na tej liście, hello world. Aplikacja Hello World jest zdecydowanie najczęstszym punktem wyjścia dla każdego programisty newbie na świecie. Istnieją dosłownie miliony różnych kodów aplikacji "Hello World!", Napisanych przez tak samo wielu początkujących programistów. Wprowadzenie do kodu aplikacji Hello World w języku programowania JAVA. Istnieją trzy główne komponenty, które składają się na aplikację znaną jako "Hello World!" Te komponenty składają się z głównej metody HelloWorldApp, definicji klasy i komentarzy kodu źródłowego. O ile nie wiesz już, jak programować i wykorzystywać polecenia w JAVA, dalsze implikacje "Hello World!" Będą zrozumiałe dopiero po przeczytaniu reszty materiału programistycznego "Hello World!" w tym tekście. Poniżej słowo kluczowe "class" (pod pierwszą strzałką w pierwszym wierszu) uruchamia tak zwaną definicję klasy, która nadaje znaczenie klasie "name", a także uruchamia kod dla każdej klasy, która pojawia się pomiędzy każdym zestawem otwierających i zamykających nawiasów klamrowych na poniższym schemacie. Dla tych z was, którzy mogą nie być zaznajomieni z nawiasami klamrowymi ("{" i "}"), są one symbolem używanym do wykonywania kodu w języku JAVA. W języku JAVA każda aplikacja rozpoczyna się definicją klasy, jak zobaczycie w następnym rozdziale. Ponadto, w przypadku używania języka programowania JAVA, w każdej aplikacji musi być stosowana metoda "główna". Na koniec komentarze kodu źródłowego są przydatne tylko dla innych programistów, aby mogli lepiej zrozumieć kod. Komentarze do kodu źródłowego są całkowicie ignorowane przez kompilator programu.

Wprowadzenie do zmiennych typu Integer w języku programowania JAVA

Podczas pisania kodu z językiem programowania JAVA, każda pojedyncza zmienna musi zostać najpierw zadeklarowana, zanim będzie mogła zostać użyta w tym kodzie. Podczas pracy w języku JAVA jest to określane jako typ statyczny. Kiedy programista deklaruje zmienną, która oznacza, że nazwa zmiennej i typ zmiennej jest wyraźnie zaznaczony w kodzie. Kiedy nazwa zmiennej i jej typ są jasno określone, pozwala to programowi wiedzieć, że w tym konkretnym kodzie jest wartość początkowa. Pozwala to również programowi wiedzieć, że przechowywane są dane liczbowe i że istnieje określone pole o nazwie "yournamedobjecthere", które istnieje w programie i kodzie. Typ danych zmiennej całkowicie określa konkretne operacje, które można wykonać na it określa także wartości, które może zawierać zmienna

Wprowadzenie do używania obiektu w języku programowania JAVA

Jak stwierdzono wcześniej, JAVA jest językiem programowania, który jest bardzo silnie zorientowany obiektowo. Kiedy programista pracuje w języku programowania JAVA, wówczas programista używa obiektów w celu wykonania zadania kodu pod ręką. Programista utworzy

te obiekty, zmodyfikuje te obiekty, przesunie obiekty w obrębie programu, zmieni zmienne obiektów, wywoła metody obiektów, a następnie połączy je również z innymi obiektami w kodzie. Podczas programowania za pomocą JAVA, programista opracuje klasy w ramach kodu, programista utworzy obiekty z tych konkretnych klas w ramach tego kodu, a następnie programista użyje tych klas w kodzie z innymi klasami i innymi obiektami w kodzie. Kiedy programista pisze kod w programie JAVA, programista definiuje zestaw określonych klas w kodzie. Jak dowiesz się dalej, klasy są po prostu szablonami używanymi przez programistów do tworzenia obiektów w kodzie programisty. Obiekty, które programista chce utworzyć, które są również znane jako instancje, są samodzielnymi aspektami konkretnego programu, który ma również względne dane i funkcje. Programista w większości przypadków użyje tej klasy po prostu do tworzenia i definiowania instancji w kodzie, a następnie programisty będzie działać z tymi konkretnymi instancjami w ramach tego jednego kodu. Dlatego w części 5 dowiesz się, jak stworzyć dowolny nowy obiekt z dowolnej klasy, którą stworzyłeś. Aby stworzyć zupełnie nowy obiekt, programista musi najpierw użyć nowego operatora wraz z nazwą konkretnej klasy i ta klasa ma być używana przez programistę i IDE jako kompleksowy szablon. Pamiętaj, że podczas wykonywania nawiasów zawsze podążasz za nazwą klasy, bez względu na wszystko. Te nawiasy są bardzo ważne, więc upewnij się, że ich nie pominiesz, bo inaczej twój kod nie zostanie wykonany poprawnie lub w ogóle nie zostanie wykonany. Nawiasy po nazwie klasy mogą nawet pozostać puste. Jeśli zostawisz nawiasy po nazwie klasy puste, oznacza to, że pomyślnie utworzyłeś najbardziej podstawowy obiekt możliwy w kodzie, jak również najprostszy obiekt możliwy w twoim kodzie. Po pomyślnym utworzeniu obiektu w kodzie, prawdopodobnie zdecydujesz się użyć tego obiektu podczas pisania programu. Kiedy jako programista zdecydujesz, w jaki sposób chcesz użyć swojego obiektu, musisz wywołać jedną z metod obiektu, aby wykonać określone działanie, możesz użyć określonej wartości jednego z pól obiektu w obrębie linii kodu, lub możesz nawet zmienić jedno z pól obiektu łącznie. Wprowadzenie do używania klasy w języku programowania JAVA. Klasa jest po prostu prostym szablonem lub prostym schematem, gdy programista tworzy kilka różnych obiektów. Ten prosty szablon lub prosty schemat będzie działał w celu zdefiniowania zachowań klasy, a także jej właściwości. Obiekty klasy języka programowania JAVA pokazują zachowania, a także właściwości zdefiniowane przez konkretną klasę obiektu. Klasa w języku programowania JAVA może zawierać metody, jak również pola, aby opisać, jak obiekt powinien zachowywać się w linii kodu i wewnątrz

Wprowadzenie do używania typu danych łańcuchowych w języku programowania JAVA

Pomimo tego, że kompilator języka programowania JAVA ma szczególnie specjalne wsparcie dla String s, String sam w sobie nie jest typem danych pierwotnych, ponieważ String zawiera szeroki wachlarz znaków. W tym przypadku String jest po prostu referencyjnym typem danych i klasą w programowaniu JAVA. W celu odniesienia, przykłady obsługi String s obejmują takie rzeczy jak wykonywanie tzw. Konkatenacji String, a także programista wybierający konwersję literałów łańcuchowych - co nie jest typem pierwotnym - do instancji String, które są typem pierwotnym. Używając języka programowania JAVA, String faktycznie zapożycza ze składni programowania C. Oznacza to, że kompilator języka programowania JAVA rozumie ciąg jako coś, co jest znane jako tablica znaków. W związku z powyższym String jest tylko abstrakcyjnym

typem danych w języku JAVA, który jest faktycznie tworzony przez prymitywną tablicę znaków typu danych, typ danych zapożyczony z programu JAVA - języka programowania C.

Anatomia instrukcji if w języku JAVA

W najprostszej i najbardziej podstawowej możliwej formie instrukcji "if" wykonuje instrukcję lub blok instrukcji, jeśli wyrażenie boolowskie ma wartość true. Wartość logiczna to tylko jedna odpowiedź 1 lub 0, albo tak, albo nie, albo albo prawda, albo fałsz.

Anatomia instrukcji if-else w języku JAVA

Gdy instrukcja "if" zawiera klauzulę else, efektywnie wykonuje instrukcję, jeśli wyrażenie logiczne ma wartość true lub wysyła blok, jeśli wyrażenie boolean ma wartość false.

Anatomia pętli w języku programowania JAVA

W języku JAVA istnieją 3 różne typy pętli. Istnieje "pętla for", "pętla while" i "pętla do-while". Pętla 'for' (trafnie) wielokrotnie wykonuje blok instrukcji dopóki warunek napisany przez programistę nie zostanie zwrócony jako false. Pętla "while" wykonuje blok instrukcji, gdy wyrażenie Boolean okaże się prawdziwe, a pętla "while" zostanie zakończona, gdy to samo wyrażenie okaże się fałszywe. Pętla "wykonaj podczas gdy" jest bardzo podobna do pętli "while", z tym wyjątkiem, że przetwarza warunek binarny tylko po wykonanie bloku instrukcji. zdefiniowane boolean i Boolean Przede wszystkim wartość "Boolean" jest wartością, która ma opcję tylko jednej z dwóch opcji. Te opcje są albo "tak", albo "nie", albo 1 albo 0, albo "prawda" lub "fałsz". W języku programowania JAVA wartości logiczne mają typ zmiennej. Na przykład: boolean user = true. Zamiast wpisywać "string" lub "double" lub "int", programista po prostu wpisuje "boolean" z małą literą "b".

Jak rozpocząć pisanie programów krok po kroku za pomocą narzędzia JAVA

Ze względu na swoją elastyczność JAVA jest jednym z najczęściej używanych języków programowania używanych przez twórców oprogramowania. Jak wcześniej wspomniano w tej książce, JAVA jest podstawowym językiem programowania, który jest używany w rozwoju aplikacji dla systemu Android, a JAVA jest bardzo szeroko stosowana również w back-endowej stronie programowania. Jeśli jesteś nowicjuszem w świecie programowania i szukasz lub myślisz o zrobieniu kariery lub hobby w tej dziedzinie, poniższy przewodnik krok po kroku dla początkujących da ci dobry pomysł, czego możesz się spodziewać, ucząc się programowania i pisania twój własny kod. Ponieważ jest łatwy w użyciu i tak szeroko wykorzystywany przez wszelkiego rodzaju programy i programistów, programowanie JAVA jest doskonałą opcją dla zainteresowanych autorów kodu po raz pierwszy, nawet jeśli nie koniecznie masz na uwadze, aby Twoje cele stały się karierą w pełnym wymiarze godzin w tej dziedzinie. . Poniższy przewodnik krok po kroku dostarczy ci naprawdę solidnych podstaw na początku podstaw programowania, koncepcji programowania obiektowego lub OOP, lepszego zrozumienia informatyki, a nawet pozwoli ci znaleźć właściwą drogę do sukcesu w inżynierii oprogramowania. Ta informacja z powodzeniem pozwoli ci wyprzedzić grę, jeśli chodzi o karierę w informatyce, którą naprawdę interesujesz.

1. Jak zainstalować JAVA Development Kit lub JDK.

- * Otwórz przeglądarkę i wyszukaj JAVA JA
- * Wyszukaj opcję z witryny Oracle.com i kliknij
- * Kliknij na pobranie JAVA JDK
- * Kliknij, aby zaakceptować licencję
- * Wybierz swój system operacyjny
- * Zminimalizuj wszystko inne na komputerze
- * Po pobraniu przygotuj się do uruchomienia pliku wykonywalnego
- * Kliknij Dalej i sprawdź status
- * JAVA JDK rozpocznie instalację
- * Kliknij przycisk Dalej i poczekaj, aż JAVA zakończy instalację
- * Po zainstalowaniu możesz przejść do folderu C i powinieneś zobaczyć dwa pliki, JDK i JRE

2. Jak zainstalować i skonfigurować Eclipse Integrated Development Environment lub IDE

- * Wpisz www.eclipse.org w przeglądarce
- * Wybierz opcję pobrania Eclipse IDE dla programistów JAVA
- * Wybierz swój system operacyjny
- * Nastąpi przekierowanie, aby wybrać najbliższą lokalizację komputera, kliknij lokalizację znajdującą się najbliżej Ciebie
- * Zostaniesz poproszony o zapisanie pliku zip, zapisz plik zip
- * Otwórz plik zip po zakończeniu pobierania
- * Zminimalizuj wszystko inne na komputerze
- * Plik zip zawiera folder o nazwie eclipse, wypakuj do folderu C.
- * Po zakończeniu ekstrakcji powinieneś widzieć Eclipse w twoim folderze C.
- * Otwórz Eclipse
- * Po raz pierwszy otwierasz Eclipse, pojawi się pytanie, które zapyta Cię, gdzie chcesz zapisać swoje projekty - wybierz lokalizację lub pozostaw ją w domyślnym folderze C
- * Kliknij przycisk OK, a zaćmienie powinno wkrótce zostać otwarte
- * Aby rozpocząć zupełnie nowy projekt, przejdź do pliku, następnie kliknij nowy, a następnie kliknij projekt JAVA
- * Nazwij swój projekt
- * Zamień na pierwszą literę każdego ze słów w nazwie projektu

- *Kliknij Następny
- * Kliknij Zakończ
- * Tworzony jest folder "Mój projekt"

3. Jak stworzyć swój pierwszy kod JAVA w Eclipse IDE

- * Rozpocznij nowy projekt w ramach Eclipse
- * Zobaczysz, że Eclipse utworzy również podkatalog dla tworzonego projektu, to będzie zawierać twój plik źródłowy
- * Zamknij wszystkie projekty z wyjątkiem projektu, nad którym będziesz pracować
- * Kliknij prawym przyciskiem myszy na folderze źródłowym projektu, nad którym chcesz pracować, a będziesz mógł dodawać pakiety do swojego folderu (za każdym razem, gdy dodasz pakiet, dodasz folder do folderu źródłowego)
- * Uzyskaj dostęp do właśnie utworzonego pakietu, a będziesz mógł utworzyć nową klasę
- * Podczas nadawania nazw klasie, upewnij się, że pierwsza litera każdego wyrazu wybranej przez siebie nazwy jest wielka
- * Zostaniesz zapytany, który sposób zatrzymania metody chcesz utworzyć, sprawdź publiczny stan pustki
- * Naciśnij Zakończ
- * Będziesz chciał dołączyć do swojego projektu sekcję komentarzy; Aby to zrobić, zacznij od ukośnika, a następnie gwiazdki i naciśnij enter
- * Ta sekcja komentarzy nie jest wykonywana przez twój program, to tylko informacje, które dokumentujesz dla swojego programu
- * Innym sposobem tworzenia komentarzy jest rozpoczęcie od dwóch ukośników i umieszczenie komentarza po; to też nie zostanie wykonane
- * Cokolwiek napiszesz w głównej klasie zostanie wykonane
- * JAVA ma wbudowane klasy, które pomogą Ci programować; najważniejszym z nich jest System - upewnij się, że S jest napisane wielką literą
- * Po wpisaniu Systemu, okresowe zaćmienie automatycznie rozpozna, co należy zrobić; powinien wyglądać tak: System.
- * Pojawi się menu podręczne, pokazujące dostępne metody etykiety w tej klasie systemu
- * Wybierz metodę o nazwie out.println
- * Po wybraniu, będziesz mógł go kliknąć i zacząć dodawać komentarze

* Kiedy będziesz gotowy do uruchomienia swojego programu, klikniesz prawym przyciskiem myszy na swój projekt i uruchomisz go jako aplikację JAVA - lub - możesz kliknąć opcję "uruchom moją klasę", która otworzy okno "Zapisz i uruchom"

* Za każdym razem, gdy przewijasz wbudowaną metodę klasy w edytorze eclipse, otworzysz okno pomocy, które pokazuje, do czego ten konkretny kod może zostać użyty

4. Programowanie za pomocą zmiennych i typów w języku JAVA

* Podczas deklarowania zmiennej w JAVA musisz najpierw wybrać typ danych

* Na przykład, powiedzmy, że chcesz zapisać liczbę całkowitą w zmiennej

* Zadeklaruj typ danych jako "int" (dla liczby całkowitej)

* Nazwij zmienną, powiedz nazwę swojej zmiennej x

* Przypisz wartość do swojej zmiennej, na przykład 10

* Twój wiersz kodu powinien wyglądać teraz tak: `int x = 10;`

* Możesz również bezpośrednio użyć swojej zmiennej w linii `System.out.println`

* Zamiast wpisywać wiadomość w tym wierszu, możesz wstawić x jako wartość; twoja linia powinna wyglądać następująco:

```
System.out.println (x);
```

* Gdy ten program zostanie uruchomiony, wartość przypisana x, która wynosi 10, zostanie wydrukowana tam, gdzie x jest w linii

* Podstawowe typy danych obejmują:

1. bajt - który jest liczbą i 1 bajtem
2. krótki - który jest liczbą i 2 bajtami
3. int - który jest liczbą i 4 bajtami
4. a long - który jest liczbą i 8 bajtami
5. float - który jest liczbą zmiennoprzecinkową i 4 bajtami
6. double - który jest liczbą zmiennoprzecinkową i 8 bajtami
7. char - który jest znakiem i 2 bajtami
8. a boolean - która jest prawdą lub fałszem i 1 bajtem

* Bajt, krótko, int i długo mogą wszystkie przechowywać liczbę, jedyną różnicą jest ich wielkość - jak wskazano

* Zakres wszystkich zmiennych rośnie wraz z rozmiarem bajtów

* Krótki przykład: `short my_variable = 10;`

* W przypadku używania zmiennej zmiennoprzecinkowej do kodu należy dodać słowo float:
float my_decimal = (float) 4.5;

* W przypadku użycia podwójnej zmiennej nie jest konieczne różnicowanie i powinno wyglądać tak: double my_double = 11.52;

* Użyj char, jeśli chcesz przechowywać pojedynczy znak: char

```
my_char = "A";
```

* Boolean może przechowywać true lub false: boolean is_true = false;

* Po wydrukowaniu pojawiłyby się tylko wartości wszystkich tych zmiennych, aby były wyświetlane jako 10, 4.5, 11.52, A i false

5. Co to znaczy, aby uzyskać dane wejściowe użytkownika za pomocą JAVA

* W języku JAVA istnieje klasa o nazwie Skaner, która umożliwia otrzymywanie danych wejściowych i wygląda następująco: Skaner skan = nowy skaner (System.in);

* Aby móc podawać dane wyjściowe w tej samej linii, powinno wyglądać to następująco: Skaner skanera = nowy skaner (System.out);

* Jeśli chcesz zebrać dane wejściowe od użytkownika, Twój pierwszy wiersz kodu powinien wyglądać następująco: System.out.println ("Wprowadź numer");

* Następnie chcesz nazwać zmienną i kontynuować kodowanie:

```
int user_input_number = skanuj.
```

* Gdy dojdiesz do części w drugim wierszu podczas skanowania., Eclipse poprosi cię o dokonanie wyboru, tutaj chcesz wybrać następny int, więc ukończona druga linia będzie wyglądać następująco: int user_input_number = scan.next int ();

* Jeśli chcesz wydrukować tę wartość, ten wiersz kodu powinien wyglądać tak: System.out.println ("Podana wartość jest");

* Jeśli chcesz wydrukować wydruk bez przerywania linii, najpierw pomiń ln po wydrukowaniu w wierszu kodu, następnie skopiuj zmienną, która w tym przypadku to user_input_number, następnie wklej ją w nawiasach i uruchom program, a następnie ostatnia linia pojawi się w ten sposób: System.out.print (user_input_number);

* Po uruchomieniu tego programu zauważysz, że prosi użytkownika o podanie wartości. Powiedzmy, że wpisujesz wartość 1000

* Po wprowadzeniu wartości program potwierdzi wprowadzoną wartość

* Double działa w ten sam sposób, ale musisz upewnić się, że konsekwentnie zmieniasz całą linię kodu, na przykład:

```
Scanner scan1 = new Scanner (pojawi się jako "System.in");
```

```
System.out.println (Wprowadź tu wartość dziesiętną);
```

- * `double user_input_double = scan1.nextdouble ();`
- * `System.out.println ("Podana wartość jest");`
- * `System.out.print (user_input_double);`
- * Jeśli chcesz wprowadzić tekst, musisz zdefiniować zmienną jako ciąg
- * Weź poprzedni kod i po prostu zmień linię `scan.next`, ponieważ oczekujesz linii od użytkownika
- * Cały program powinien wyglądać następująco:

Pierwsza linia: `"Scanner scan1 = new Scanner (System.in);"`

Druga linia: `"System.out.println (" Podaj ciąg znaków ");"`

Trzecia linia: `"string user_input_string = scan1.nextLine ();"`

Czwarta linia: `"System.out.println (" Wprowadzony łańcuch znaków ");"`

Piąta linia: `"System.out.print (user_input_string);"`

- * Gdy uruchomisz ten program, zostaniesz poproszony o wpisanie jakiegoś ciągu znaków, wprowadź wybraną frazę

6. Zapoznanie się z operacjami matematycznymi i arytmetycznymi w języku JAVA

- * Najpierw zadeklaruj dwie zmienne `x` i `y` w tym przypadku
- * Następnie przypisz odpowiedź
- * `int x, y, odpowiedź;`
- * Przypisz wartość do `x` i `y`, odpowiednio 20 i 30
- * Przydziel wartość do `odpowiedzi`, 50
- * Jeśli chcesz wydrukować odpowiedź, program powinien wyglądać tak:

`int x, y, odpowiedź;`

`x = 20;`

`y = 30;`

`odpowiedź = x + y;`

`System.out.println ("Answer =" + answer);`

- * Gdy symbol dodawania lub plus jest używany w funkcji drukowania, jest znany jako operator konkatenacji lub konkatenacji
- * Poprawna odpowiedź powinna zostać wydrukowana w czasie zaćmienia bez faktycznego wykonywania matematyki, o ile kod jest poprawny

- * Aby dokonać odejmowania, wystarczy tylko zamienić symbol dodawania na symbol odejmowania w czwartym wierszu kodu lub linię odpowiedzi: `answer = x - y;`
- * Podczas wykonywania mnożenia `x` nie może być użyty jako symbol mnożenia, ponieważ `x` jest często zmienną; zamiast tego używana jest gwiazdka
- * Wszystko, co musisz zrobić, aby wykonać mnożenie, to ponowne zastąpienie symbolu: `answer = x * y;`
- * Podział jest nieco trudniejszy - jeśli podzielisz niższą liczbę o wyższą liczbę i zadeklarujesz swoją zmienną jako liczbę całkowitą, odpowiedź zawsze będzie wynosić zero, ponieważ liczby całkowite nie mogą być dziesiętne
- * W tym samym świetle, gdy zmienna została zadeklarowana jako liczba całkowita i dzielisz wyższą liczbę, która nie jest podzielna przez niższą liczbę bez użycia liczb dziesiętnych, również nie otrzymasz dokładnie matematycznie poprawnej odpowiedzi - jeśli spróbujesz podzielić siedemdziesiąt do trzydziestu, twoja odpowiedź będzie dwójka, ponieważ tyle razy trzydzieści może przejść na siedemdziesiąt, a cała liczba jako odpowiedź
- * Aby uzyskać dokładne odpowiedzi w podziale, musisz zadeklarować swoje zmienne jako podwójne
- * Symbol przekroju poprzecznego jest używany zamiast innych symboli podczas dzielenia
- * Jeszcze jeden operator w JAVA nazywa się operator modułus i daje resztę podziału

7. Zapoznanie się z operatorem inkrementacji i operatorem przypisania

- * Operator inkrementacji jest używany, gdy chcesz zwiększyć wartość zmiennej i nieco przypomina algebrę
- * `int x = 10;`
- * `x = x + 1;`
- * `System.out.println (x);`
- * Innym sposobem napisania tego kodu w celu osiągnięcia tego samego celu byłoby zastąpienie `+ 1` dwoma symbolami dodawania, jest to tak zwana operacja przyrostu postu
- * `int x = 10;`
- * `x ++;`
- * `System.out.println (x);`
- * Ta sama odpowiedź jedenastu została osiągnięta zarówno w druku
- * Jeśli użyjesz `x ++` w linii drukowania, wygląda to następująco:
- * `int x = 10;`
- * `System.out.println (x ++);`

* W operacji przyrostu postu wartość x zmieni się tylko po tej operacji plus

* Teraz zwiększysz wartość x o jeden, który powinien wyglądać tak:

```
* int x = 10;
```

```
* System.out.println (x ++);
```

```
* System.out.println (x);
```

* Przed operacją przyrostu symbole plus umieszczane są przed znakiem x i zwiększają jego wartość przed wykonaniem operacji

* Więc jeśli dodamy ++ x zamiast x ++ w poprzednim kodzie, zobaczysz:

```
* int x = 10;
```

```
* System.out.println (++ x);
```

```
* System.out.println (x);
```

* Odpowiedź brzmi:

```
* 11
```

```
* 11
```

* Wynika to z tego, że operator prekrementacji, w którym x miał wartość 11, i operator przyrostu postów, gdzie x również miał wartość 11, zostały skompilowane

* Teraz powiedzmy, że chcesz dodać pięć do tej wartości, możesz to zrobić

wypisując: `x = x + 5`

* Innym sposobem zapisu jest: `x + = 5`

* Jeśli chcesz pomnożyć wartość 5 i wartość 10 w równaniu, wystarczy zastąpić symbol dodawania w `x + = 5` gwiazdką - ten skrót również działa na mnożenie - i będzie wyglądał jak to: `x * = 5`

* Termin "równy" w języku JAVA jest określany jako operator przypisania

* Zatem `x * = 5` i `x = x * 5` jest operatorem przypisania, a ten operator pracuje nad dodawaniem, odejmowaniem, mnożeniem i dzieleniem

8. Jak włączyć instrukcje else lub warunkowe i ich operatory relacyjne

* Instrukcja warunkowa jest instrukcją, która określa, czy warunek jest prawdziwy, czy fałszywy, i na podstawie tego warunku wykonuje określony kod

* Podwójny znak równości `==` symbolizuje operator relacyjny

* Instrukcja if wygląda następująco:

```
int x = 10;
```

```
jeśli (x == 10) {
```

```
System.out.println ("yes x == 10");
```

Ta instrukcja mówi, że jeśli x jest równe 10, to chcę uruchomić ten kod. Tutaj program odpowiada, że tak, x jest w rzeczywistości równe 10, co jest prawdą.

* Oznacza to, że instrukcja jest przestrzegana i warunek zostanie wykonany

* Teraz spróbuj sprawdzić, czy x jest równe 20, zmieniając tylko instrukcję, umieszczając liczbę 20 w wierszu if

* Żadna odpowiedź nie pojawi się w czasie zaćmienia, ponieważ ta instrukcja jest fałszywa i warunek nie może zostać wykonany

* Instrukcja if może wykonywać się, gdy instrukcja if nie działa i wygląda następująco:

```
* int x = 10;
```

```
* jeśli (x == 20) {
```

```
* System.out.println ("tak x == 10");
```

```
* }
```

```
* jeszcze {
```

```
* System.out.println ("bez x! = 10");
```

* Zwróć uwagę na wykrzyknik, który zajął miejsce jednego z równych znaków za linią else - nazywa się to operatorem nierówności

* Teraz gdy ponownie uruchomisz program, eclipse da ci odpowiedź, że x w rzeczywistości nie jest równy 10

* W JAVA operatory porównania obejmują:

1. == jest równe

2. != Nie jest równe

3. > jest większy niż

4. < jest mniejsze niż

5. >= jest większe lub równe

6. <= jest mniejsze lub równe

9. Jak wykorzystać logiczne operatory w JAVA

* Jeśli chcesz ocenić więcej niż jeden warunek za pomocą instrukcji if lub instrukcji if, musisz użyć operatora logicznego

* Istnieją dwa podstawowe rodzaje operatorów logicznych - operator i, który jest symbolizowany przez podwójne ampersands &&, a drugi typ nazywany jest operatorem symbolizowanym przez symbole podwójnych potoków ||

* Operator i operator lub operator mogą jednocześnie używać dwóch warunków jednocześnie w instrukcjach if i if else

* Operator i sprawdza, czy wszystkie warunki są jednocześnie prawdziwe, podczas gdy operator lub sprawdza, czy jedna lub więcej rzeczy jest prawdziwa

* Symbol umieszczony pomiędzy obiektami mówi, którego używasz, jak w

wiersz czwarty z następujących dwóch kodów:

```
int subject1 = 40;
```

```
int subject2 = 60;
```

```
// && -> AND || -> OR
```

```
if ((subject1 >= 35) && (subject2 >= 35)) {
```

```
System.out.println ("warunek jest prawdziwy");
```

Ten warunek jest prawdziwy, ponieważ zarówno 40, jak i 60 są większe niż 35.

* Teraz spróbuj dodać instrukcję else za pomocą instrukcji if, nakrętka najpierw zmieni wartość podmiotu od 1 do 20

Jeśli inaczej

```
int subject1 = 20;
```

```
int subject2 = 60;
```

```
// && -> AND || -> OR
```

```
if ((subject1 >= 35) && (subject2 >= 35)) {
```

```
System.out.println ("warunek jest prawdziwy");
```

```
} else {
```

```
System.out.println ("warunek jest fałszywy");
```

Ten warunek jest fałszywy.

* Teraz spróbuj uruchomić ten sam program, ale zamień operatora i na operatora:

```
int subject1 = 40;
```

```
int subject2 = 60;
```

```
// && -> AND || -> OR
```

```
if ((subject1 >= 35) || (subject2 >= 35)) {
```

```
System.out.println ("warunek jest prawdziwy");
```

```
} else {
```

```
System.out.println ("warunek jest fałszywy");
```

Ten warunek jest prawdziwy

10. Praca z instrukcją przełącznika w JAVA

* Kiedy będziesz musiał sprawdzić wielokrotność w JAVA, będziesz chciał

użyć instrukcji switch warunków

* Możesz także zmienić instrukcje zamiast instrukcji else

* Przerwy w instrukcji switch mają na celu rozbić odpowiedzi, które otrzymasz z programu po uruchomieniu

* Część instrukcji switch zawiera użycie 'default', który działa tak samo jak else w instrukcji if else

* int = 90;

// bajt, int, short lub char.

przełącznik (wynik)

```
{case 90:
```

```
System.out.println ("Very good");
```

```
złamać;
```

```
sprawa 60:
```

```
System.out.println ("Good");
```

```
złamać;
```

```
sprawa 30:
```

```
System.out.println ("OK");
```

```
złamać;
```

```
domyślna :
```

```
System.out.println ("Oceny nie są zdefiniowane");
```

```
złamać;
```

```
}
```

Przy wartości liczby całkowitej określonej jako 90, instrukcja case 90 jest prawdziwa.

* Gdy nie ma przerwy między instrukcjami case, program domyślnie wybierze następny najniższy z rzędu podczas korzystania z tego typu kodu

11. Praca z instrukcjami while w języku JAVA, zwanymi także pętlami while

* Podczas gdy pętle są najbardziej podstawowymi pętlami w JAVA

* A jest fragmentem kodu lub instrukcją, która wykonuje pewien blok kodu raz po raz, dopóki nie zostanie spełniony jakiś warunek

* Jeśli chcesz wielokrotnie wykonywać kod bez konieczności przepisywania go w kółko, to jest to, do czego służy pętla:

```
int a = 0;

while (a <= 10)

{

System.out.println (a);

a ++;

}
```

12. Przechodzenie do instrukcji do wykonania w języku JAVA, znanych również jako pętle do while

* Podstawowa odmiennosc od pętli while do pętli do while polega na tym, że chociaż pętle najpierw oceniają warunek, a następnie wykonują kod, pętle do while wykonują najpierw kod, a następnie oceniają warunek

13. Jak nauczyć się implementować tablice w swoim programie JAVA

* Tablica jest podobna do zmiennej, ale może przechowywać więcej niż jedną wartość naraz - jedynym warunkiem jest to, że nawet jeśli możesz przechowywać więcej niż jedną wartość w tablicy, musi to być ten sam typ wartości

* Na przykład, możesz zapisać 10 liczb całkowitych w tablicy, ale jeśli chcesz przechowywać 5 liczb całkowitych i 5 podwójnych, nie będziesz w stanie tego zrobić

* Jest to jeden sposób deklarowania tablicy: `int [] myIntArray = {4, 2,1,5, 3};`

* Inny sposób to: `int myIntArray2 [] = {4,2,1,5, 3}` Ta metoda jest dobra, ale nie jest preferowanym sposobem deklarowania tablicy

* Istnieją jeszcze trzy sposoby deklarowania tablic:

1. `int [] myIntArray = new int [3];`

2. `int [] myIntArray = {1,2,3};`

3. `int [] myIntArray = new int [] {1,2,3};`

14. Jak używać łańcucha JAVA w twoim programie

- * Ciąg znaków jest sekwencją znaków i może być również znany jako tablica znaków
- * W celu zadeklarowania ciągu znaków w języku JAVA wystarczy użyć słowa kluczowego "ciąg"
- * Następnie nazwij swój ciąg, jak tylko chcesz
- * Następnie umieszczasz znak równości
- * Następnie w podwójnych cudzysłowach umieść dowolny ciąg, który chcesz przypisać i wydrukuj go tak samo, jak wszystkie inne:

```
String mystring = "Hello World";
```

```
System.out.println (mystring)
```

15. Wprowadzenie do stosowania metod w programowaniu JAVA * Terminy metoda i funkcja mogą być używane w sposób zamienny, ale preferowanym terminem w JAVA jest metoda

- * Metoda jest kawałkiem kodu, który wykonuje pewną logikę i możesz zawinąć tę metodę z nazwą i możesz przywołać tę metodę tyle razy, ile chcesz, gdy chcesz jej użyć
- * Możesz nazwać swoją metodę dowolnie
- * Jest to najbardziej podstawowy rodzaj metody:

```
public static void myFirstMethod () {  
}
```

- * Termin "public static" jest znany jako smoczek

Jeśli chcesz wydrukować wiadomość za pomocą metody, jest to składnia lub

kod wyglądałby tak:

```
public static void myFirstMethod () {  
System.out.println ("Hello Youtube");  
}
```