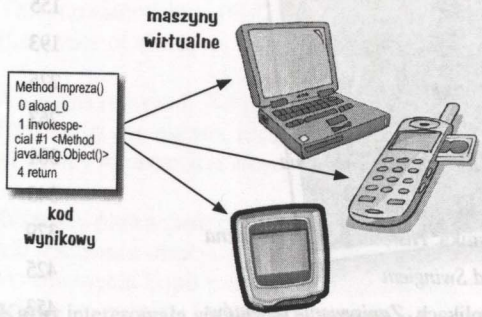
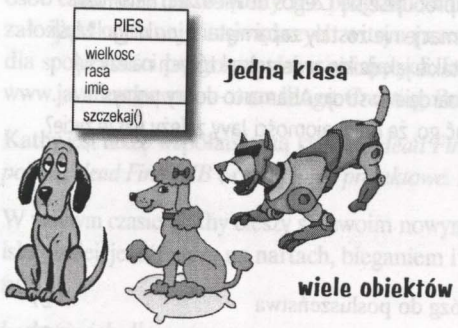
1 Przełamując zalew początkowych trudności

Java zabiera nas w nowe miejsca. Od momentu pojawienia się pierwszej, skromnej wersji o numerze 1.02 Java pociągała programistów ze względu na przyjazną składnię, cechy obiektowe, zarządzanie pamięcią, a przede wszystkim obietnicę przenośności. Od samego początku „wskoczymy na głęboką wodę" — napiszemy prosty program, skompilujemy go i wykonamy. Porozmawiamy o składni, pętlach, rozgałęzieniach oraz innych czynnikach, które sprawiają, że Java jest taka fajna. Zatem wskakuj!



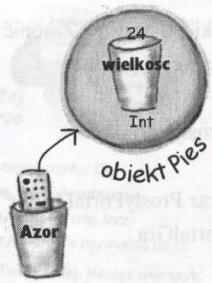
2 Wycieczka do Obiektowa

Mówiono mi, że będą tam same obiekty. W rozdziale 1. cały kod programu został umieszczony w metodzie `main()`. Nie jest to w pełni zgodne z zasadami programowania obiektowego. A zatem trzeba porzucić świat programowania proceduralnego i rozpocząć tworzenie własnych obiektów. Zobaczymy, co sprawia, że programowanie obiektowe w Javie jest takie fajne. Wyjaśnimy także, na czym polega różnica pomiędzy klasą a obiektem. W końcu pokażemy, w jaki sposób obiekty mogą ułatwić nam życie.



3 Poznaj swoje zmienne

Istnieją dwa rodzaje zmiennych: zmienne typów podstawowych oraz odwołania. W programach będą istnieć także inne typy danych niż jedynie liczby całkowite, łańcuchy znaków i tablice. Co zrobić, jeśli chcielibyśmy stworzyć obiekt **`WłaścicielZwierzaka`** zawierający składową **`Pies`**? Albo **`Pojazd`** ze składową **`Silnik`**? W tym rozdziale ujawnimy tajemnice typów danych w Javie oraz pokażemy, co można *zadeklarować* jako zmienną, *zapisać* w zmiennej oraz co z taką zmienną można potem zrobić. W końcu przekonamy się, jak wygląda życie na automatycznie porządkowanej stercie.

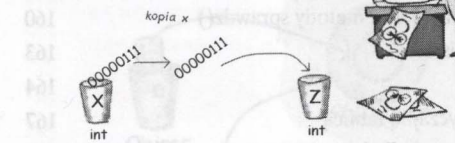


odwołanie do obiektu Pies

4 Jak działają obiekty?

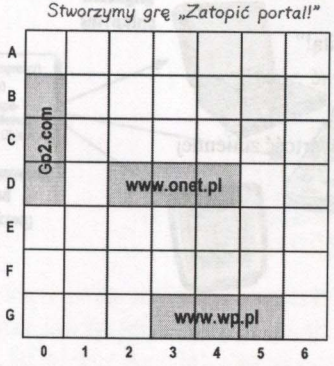
Stan wpływa na działanie, a działanie wpływa na stan. Wiemy, że obiekty mają swój **stan** i określone **działanie** reprezentowane odpowiednio przez **składowe** i **metody**. Teraz sprawdzimy, w jaki sposób stan i działanie obiektów są ze sobą *powiązane*. Otóż w swych działaniach obiekty wykorzystują swój unikalny stan. Innymi słowy, ***metody wykorzystują wartości składowych obiektów***. Na przykład: „Jeśli pies waży mniej niż 20 kilogramów, szczekamy radośnie, w przeciwnym przypadku...". ***Spróbujmy zmienić stan obiektu!***

Java przekazuje argumenty przez wartość.
To oznacza, że przekazywana jest kopia.



5 Supermocne metody

Dodajmy naszym metodom nieco siły. Zajmowaliśmy się już zmiennymi, eksperymentowaliśmy z kilkoma obiektami i napisaliśmy kod paru programów. Jednak potrzeba nam więcej narzędzi. Na przykład **operatorów**. I **pętli**. Może przydałoby się **wygenerować kilka liczb losowych** i **zamienić łańcuch znaków na liczbę całkowitą**. O tak, to byłoby super. I dlaczego nie nauczyć się tego wszystkiego, *tworząc* coś rzeczywistego, by przekonać się, jak wygląda pisanie (i testowanie) programu w Javie od samego początku do końca. **Może jakąś grę**, taką jak „Zatopić portal" (podobną do gry w okręty).



6 Korzystanie z biblioteki Javy

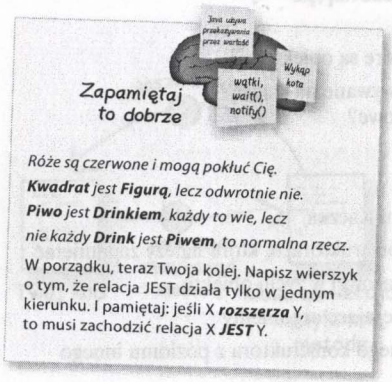
Java jest wyposażona w setki gotowych klas. Jeśli tylko dowiesz się, jak znaleźć w bibliotece Javy (nazywanej także **Java API**) to, czego szukasz, nie będziesz musiał ponownie wymyślać koła. *W końcu masz ciekawsze rzeczy do roboty.* Jeśli masz zamiar napisać program, to równie dobrze możesz napisać *tylko* te jego fragmenty, które są unikalne. Standardowa biblioteka Javy to gigantyczny zbiór klas, które tylko czekają, aby użyć ich jako klocków przy tworzeniu programów.

„Dobrze jest wiedzieć, że w pakiecie java.util istnieje klasa ArrayList. Ale jak mogłabym się o tym sama dowiedzieć?"

— Julia, lat 31, modelka

7 Wygodniejsze życie w Obiektowie

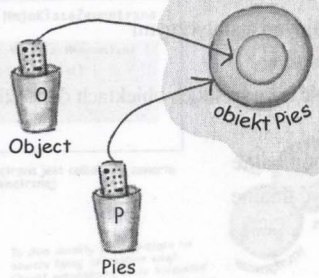
Planuj swoje programy, myśląc o przyszłości. A gdyby tak można tworzyć kod, który *inni* programiści mogliby rozbudowywać, i **to w prosty sposób**? A gdyby tak można było tworzyć elastyczny kod z myślą o tych najnowszych zmianach wprowadzanych w specyfikacji? Kiedy dowiesz się o Planie Polimorfizmu, poznasz 5 kroków służących projektowaniu lepszych klas, 3 sztuczki z polimorfizmem oraz 8 sposobów tworzenia elastycznego kodu, a jeśli zaczniesz już teraz, dodatkowo otrzymasz 4 podpowiedzi odnośnie do wykorzystywania dziedziczenia.



8 Poważny polimorfizm

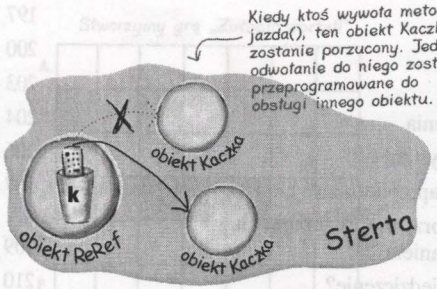
Dziedziczenie to tylko początek. Aby w pełni wykorzystać polimorfizm, niezbędne nam będą interfejsy. Musimy pójść dalej, poza proste dziedziczenie, i uzyskać elastyczność, jaką może zapewnić wyłącznie projektowanie i kodowanie z wykorzystaniem interfejsów. Czym jest interfejs? W 100% abstrakcyjną klasą. A czym jest klasa abstrakcyjna? To klasa, która nie pozwala na tworzenie obiektów. A co nam po takiej klasie? Przeczytaj, to się dowiesz...

```
Object o = lista.get(indeks);
Pies p = (Pies) o;
p.szczekaj();
```



9 Życie i śmierć obiektu

Obiekty się rodzą i umierają. To Ty jesteś za to odpowiedzialny. To Ty decydujesz, kiedy i jak *skonstruować* obiekt. Ty także decydujesz, kiedy go *porzucić*. **Odśmiecacz pamięci** odzyska pamięć zajmowaną przez niepotrzebne obiekty. Przyjrzymy się, w jaki sposób obiekty są tworzone, gdzie „żyją" i w jaki sposób efektywnie je przechowywać lub porzucać. Oznacza to, że będziemy dyskutować o stercie, stosie, zasięgach, konstruktorach, konstruktorach nadrzędnych, odwołaniach pustych oraz przydatności odśmiecacza pamięci.



W składowej „k" jest zapisywany nowy obiekt Kaczka, przez co oryginalny (pierwszy) obiekt zostaje porzucony. Teraz ten pierwszy obiekt do niczego się nam już nie przyda.

10 Liczby mają znaczenie

Zabaw się w matematyka. Java API udostępnia metody do wyznaczania wartości bezwzględnej, zaokrąglania liczb, określania wartości minimalnej i maksymalnej i tak dalej. A co z formatowaniem? Moglibyśmy chcieć wyświetlać liczby ze znakiem dolara na początku i dwoma miejscami dziesiętnymi. Albo wyświetlić daną w formie daty. Albo daty w zapisie stosowanym w Anglii. A co z przekształcaniem łańcucha znaków na liczbę? Lub zamianą liczby na łańcuch znaków? Zaczniemy jednak od wyjaśnienia, co oznacza, że zmienne lub właściwości są *statyczne*.

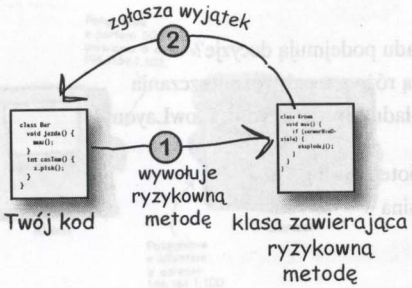
Zmienne statyczne są współdzielone przez wszystkie kopie klasy

Składowe — po jednej w każdym obiekcie.

Składowe statyczne — jedna dla całej klasy.

11 Ryzykowne działania

Czasami zdarzają się nieprzewidziane sytuacje. Pliku nie ma tam, gdzie powinien być. Serwer został wyłączony. Niezależnie od tego, jak dobrym jesteś programistą, nie jesteś w stanie kontrolować *wszystkiego*. Kiedy tworzysz metodę, której działanie jest opatrzone ryzykiem niepowodzenia, musisz także stworzyć kod, który obsłuży potencjalnie niebezpieczną sytuację. Ale skąd wiadomo, że metoda może nieść ze sobą potencjalne niebezpieczeństwo? Gdzie umieszczać kod *obsługujący* **wyjątkowe** sytuacje? W *tym* rozdziale stworzymy odtwarzacz MIDI wykorzystujący ryzykowny interfejs programistyczny JavaSound, zatem lepiej dowiedzmy się, jak zabezpieczyć się przed potencjalnymi niebezpieczeństwami.



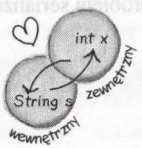
12 Historia bardzo graficzna

Musisz się z tym pogodzić — tworzenie interfejsów graficznych jest konieczne. Nawet jeśli wierzysz, że już do końca życia będziesz pisać programy działające na serwerze, to jednak będziesz także musiał pisać programy narzędziowe i zapewne będziesz chciał, aby miały one jakiś interfejs graficzny. Zagadnieniom interfejsu graficznego poświęcimy dwa rozdziały, w których przedstawimy także inne cechy języka, takie jak obsługa zdarzeń oraz klasy wewnętrzne. Naciśniemy przycisk na ekranie, wskażemy coś myszą, wyświetlimy obraz zapisany w formacie JPEG, a nawet stworzymy małą animację.

```
class MojaKlasaZewnetrzna {

  class MojaKlasaWewnetrzna {
    void doDziela() {
    }
  }
}
```

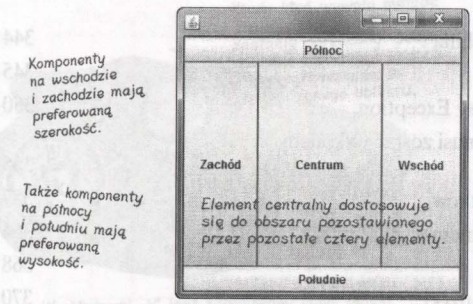
Klasa wewnętrzna jest całkowicie zawarta w klasie zewnętrznej



Te dwa obiekty przebywające na stercie łączy szczególna więź. Obiekt wewnętrzny może korzystać ze składowych obiektu zewnętrznego (i na odwrót).

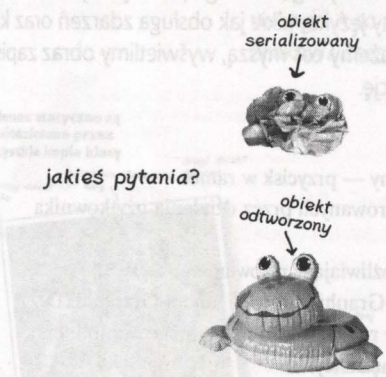
13 Popracuj nad Swingiem

Swing jest łatwy. Chyba że naprawdę zwracasz uwagę na to, co jest gdzie wyświetlane. Kod wykorzystujący Swing *wydaje się* prosty, ale kiedy go skompilujesz, uruchomisz i popatrzysz na wyniki, to często będziesz mógł sobie pomyśleć: „Hej, przecież to nie miało być wyświetlone w *tym* miejscu". To, co zapewnia prostotę kodu, sprawia jednocześnie, że trudne jest kontrolowanie położenia elementów — tym „czymś" jest **menedżer układu**. Jednak przy odrobinie pracy można nagiąć menedżer układu do naszej woli. W tym rozdziale popracujemy nad naszym Swingiem i dowiemy się czegoś więcej o różnych elementach graficznych.



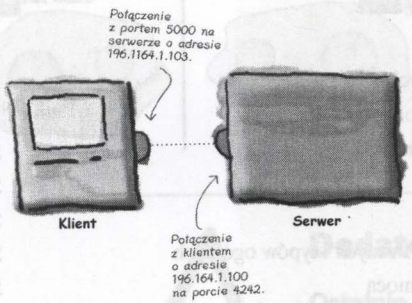
14 Zapisywanie obiektów

Obiekty można pakować i odtwarzać. Obiekty mają swój stan i działanie. Działanie obiektu określa jego klasa, natomiast *stan* zależy od konkretnego *obiektu*. Jeśli program musi zapisać stan obiektu, *możesz to zrobić w sposób trudny* — analizując każdy obiekt i pracowicie zapisując wartości wszystkich właściwości. **Możesz także zapisać obiekt w sposób łatwy i obiektowy** — „zamrażając" obiekt (przeprowadzając jego serializację), a następnie odtwarzając (poprzez jego deserializację).



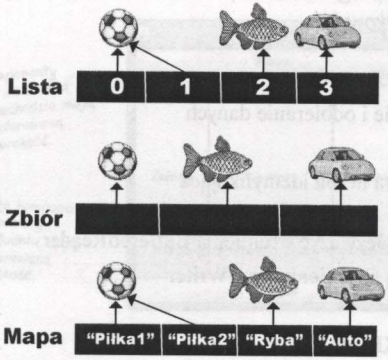
15 Nawiąż połączenie

Nawiąż połączenie ze światem zewnętrznym. To takie łatwe. Wszelkie szczegóły związane z komunikacją sieciową na niskim poziomie są obsługiwane przez klasy należące do biblioteki java.net. Jedną z najlepszych cech Javy jest to, iż wysyłanie i odbieranie danych przez sieć to w zasadzie normalne operacje wejścia-wyjścia, z tą różnicą, że są w nich wykorzystywane nieco inne strumienie. W tym rozdziale stworzymy gniazda używane przez programy klienckie. Stworzymy także gniazda używane przez programy pełniące funkcje serwerów. Napiszemy zarówno program klienta, jak i serwera. Zanim skończysz czytać ten rozdział, będziesz już dysponować w pełni funkcjonalnym, wielowątkowym programem do prowadzenia internetowych pogawędek. Zaraz... czy właśnie padło słowo *wielowątkowy*?



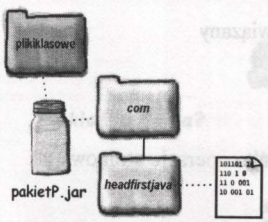
16 Struktury danych

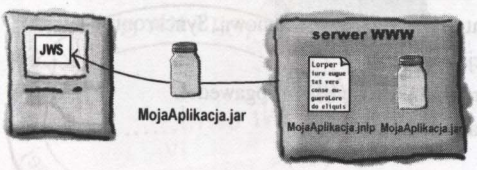
Sortowanie w Javie to pestka. Dysponujesz wszystkimi narzędziami do gromadzenia danych i manipulowania nimi, i to bez konieczności pisania własnych algorytmów. Biblioteka kolekcji Javy (*Java Collections Framework*) posiada struktury danych, które powinny sprostać praktycznie wszystkiemu, co będziesz chciał zrobić. Potrzebujesz listy, do której będziesz mógł łatwo dodawać elementy? Chcesz znaleźć coś na podstawie nazwy? A może chcesz stworzyć listę, która automatycznie będzie usuwać powtarzające się elementy? Albo posortować listę pracowników na podstawie liczby „przyjacielskich klepnięć w plecy"?



17 Rozpowszechnij swój kod

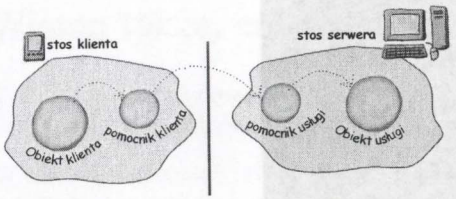
Już czas wypłynąć na „szerokie wody". Napisałeś kod swojej aplikacji. Przetestowałeś go. Udoskonaliłeś. Powiedziałeś wszystkim znajomym, że świetnie by było, gdybyś już w życiu nie musiał napisać choćby jednej linijki kodu. Ale w końcu stworzyłeś dzieło sztuki. Przecież Twoja aplikacja działa! W ostatnich dwóch rozdziałach książki zajmiemy się zagadnieniami związanymi z organizowaniem, pakowaniem i wdrażaniem kodu. Przyjrzymy się możliwościom wdrażania lokalnego, mieszanego i zdalnego, w tym wykonywalnym plikom JAR, technologii Java Web Start, RMI oraz serwletom. Nie stresuj się! Niektóre z najbardziej niesamowitych rozwiązań, jakimi może się poszczycić Java, są prostsze, niż można by przypuszczać.





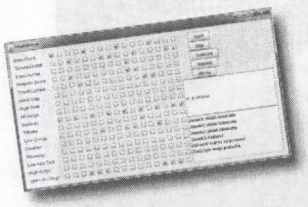
18 Przetwarzanie rozproszone

Zdalne wykonywanie aplikacji nie zawsze jest złe. Oczywiście to prawda, że życie *jest* łatwiejsze, gdy wszystkie elementy aplikacji znajdują się w jednym miejscu i są zarządzane przez jedną wirtualną maszynę Javy. Jednak takie rozwiązanie nie zawsze jest możliwe. Co w sytuacji, gdy aplikacja realizuje bardzo złożone obliczenia? Co, jeśli potrzebuje informacji pochodzących z zabezpieczonej bazy danych? W tym rozdziale przedstawiona zostanie zadziwiająco prosta technologia wywoływania zdalnych metod — *RMI* (ang. *Remote Method Invocation*). Pobieżnie przyjrzymy się także innym rozwiązaniom — serwletom, komponentom *EJB* (ang. *Enterprise Java Bean*) oraz technologii Jini.



A Dodatek A

Ostatnie doprawianie kodu. Kompletny kod aplikacji `MuzMachina` w ostatecznej wersji klient-serwer z możliwością prowadzenia muzycznych pogawędek. Twoja szansa, by zostać gwiazdą rocka.



B Dodatek B

Dziesięć najważniejszych zagadnień, które niema znalazły się w tej książce... Jeszcze nie możemy Cię zostawić i wypuścić w świat. Mamy da Ciebie kilka dodatkowych informacji, jednak to już *jest* koniec tej książki. I tym razem mówimy to zupełnie serio.

S Skorowidz