Moduł szkoleniowy Java

poziom podstawowy 15 godzinny

Człowiek - najlepsza inwestycja

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Spis treści

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

1 Metryka dokumentu

Szczeciński Park Naukowo Technologiczny			
Dokument	Moduł Szkoleniowy Java		
Krótki opis dokumentu	Moduł szkoleniowy kursu Java zawierający opis celów, treści poszczególnych lekcji, ćwiczeń oraz test końcowy sprawdzający wiedzę		
Data druku	12.02.2013	Liczba stron	110
Nazwa pliku	Java - Moduł szkoleniowy.docx	Status	roboczy

Historia zmian

Nr wersji	Data	Opis	Działanie (*)	Autorzy
0.1	01.02.2013	Utworzenie nowego dokumentu	N	Aleksander Gralak, DailyGroup Sp. z o.o.
1.0	12.02.2013	Wersja gotowa do publikacji	Z	Aleksander Gralak, DailyGroup Sp. z o.o.

(*) Działanie: N-Nowy, Z-Zmiana, W-Weryfikacja

Lista dystrybucyjna

Imię i nazwisko / Rola	Organizacja

Zgłoszono do odbioru

Imię i nazwisko		Data:		Podpis:	
Imię i nazwisko	Aleksander Gralak	Data:	12.02.2013	Podpis:	

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

2 Cel

Poznanie zastosowania i zalet języka programowania Java. Umiejętność wykorzystania podstawowych możliwości języka. Praktyczne wykorzystanie elementów środowiska pracy. Podstawowa umiejętność projektowania aplikacji zorientowanej obiektowo. Tworzenie wieloplatformowych aplikacji z interfejsem graficznym. Zwiększenie efektywności pracy poprzez umiejętność korzystania i tworzenia z dokumentacji.

3 Opis sposobu realizacji celów

10 półtoragodzinnych lekcji składających się z treści teoretycznych wraz z ćwiczeniami. Praktyczne uwagi dla osób początkujących dotyczące środowiska pracy Eclipse. Po zakończeniu całego cyklu - przeprowadzenie egzaminu sprawdzającego wiedzę.

4 Treści kształcenia

Treść kursu została podzielona na 10 bloków tematycznych – po jednym do każdej lekcji:

1. Wstęp do Javy i środowiska pracy,
2. Typy podstawowe i instrukcje sterujące,
3. Wstęp do programowania obiektowego ,
4. Interfejsy, klasy abstrakcyjne i dziedziczenie,
5. Polimorfizm, instrukcja „final”, klasy wewnętrzne i anonimowe,
6. Typ wyliczeniowy, JavaDocs ,
7. Tablice jednowymiarowe i wielowymiarowe, kolekcje,
8. Operacje wejścia – wyjścia i wyjątki ,
9. Graficzny interfejs użytkownika - Swing ,
10. Przechwytywanie zdarzeń, tworzenie plików wykonywalnych.

5 Opis założonych osiągnięć ucznia

Uczeń po odbyciu kursu pozna teoretyczne podstawy Języka Java, będzie umiał zaprojektować prostą aplikację zorientowaną obiektowo z interfejsem graficznym.

6 Sposoby osiągania celów

Po odbyciu każdej lekcji uczeń powinien samodzielnie wykonać proste ćwiczenia w celu dokładnego zrozumienia omawianych mechanizmów. Programowanie w języku Java bez odpowiedniego środowiska jest bardzo trudne dlatego jednym z celów jest jego poznanie. Umożliwi to wykonywanie ćwiczeń w czasie efektywnym.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

7 Propozycje kryteriów oceny i metod sprawdzania osiągnięć ucznia

Ocena końcowa powinna zależeć od wyniku testu sprawdzającego wiadomości teoretyczne (20 pytań zamkniętych pojedynczego wyboru - 20 punktów).

Propozycje kryteriów oceny:

- 0-9 punktów - ocena 1,
- 10-12 punktów - ocena 2,
- 13-15 punktów - ocena 3,
- 16-18 punktów - ocena 4,
- 29-20 punktów - ocena 5, Test końcowy sprawdzający wiedzę

20 zadań na 30 minutowy test sprawdzający wiedzę. Pogrubioną czcionką zaznaczono prawidłowe odpowiedzi.

1. Java kompiluje się do:
   a. Kodu maszynowego
   b. **Kodu bajtowego**
   c. Pliku skryptowego
2. Kompilator do Javy znajduje się w pakiecie:
   a. **JDK**
   b. JRE
   c. Eclipse
3. Aby przechować wartość 100 jakiego najmniejszego typu możemy użyć:
   a. long
   b. int
   c. **short**
4. Która pętla wykona jeden krok przebiegu nawet w przypadku fałszywego warunku:
   a. for
   b. **do-while**
   c. while
5. Słowo „new” tworzy:
   a. nową klasę;
   b. **nowy obiekt**
   c. nowy typ
6. Czy kiedy w klasie występuje konstruktor „public MojaKlasa(int x)” mogę stworzyć obiekt za pomocą instrukcji „new MojaKlasa()”?
   a. tak
   b. nie
   c. **tylko wtedy gdy istnieje drugi konstruktor „public MojaKlasa()”**
7. Która ze struktur nie posiada implementacji metod:
   a. klasa

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

   b. **interfejs**
   c. klasa niewłaściwa
8. Jeżeli chcemy, aby dana metoda była widoczna tylko w bieżącym pakiecie, to jakiego typu widoczności użyjemy?
   a. private
   b. public
   c. **nie podajemy typu, co oznacza użycie widoczności package (default)**
9. Które zdanie jest nieprawdziwe:
   a. Każdy obiekt typu Integer jest również obiektem typu Object
   b. Mogę używać obiektu typu Integer tak, jak każdego innego obiektu typu Object
   c. **Mogę stworzyć obiekt, który jest jednocześnie typu Integer i String.**
10. Zaznacz prawidłowe stwierdzenie
   a. **Mogę stworzyć instancję interfejsu, pod warunkiem, że zaimplementuję jego metody**
   b. Jedyną możliwością stworzenia instancji interfejsu jest użycie słowa „implements”
   c. Mogę stworzyć instancję interfejsu tak, jak zwykłej klasy.
11. Pisanie JavaDoc rozpoczynamy od znaku:
   a. **/****
   b. //
   c. @
12. Podpowiadanie składni w środowisku Eclipse wywołujemy skrótem:
   a. „Ctrl” + „Shift” + „O”
   b. **„Ctrl” + „Spacja”**
   c. „F5”
13. Jeżeli chcemy udostępnić innym programistom wybór jednego z 5 słów, powinniśmy:
   a. **Skorzystać z typu wyliczeniowego - enum**
   b. Umieścić słowa w tablicy
   c. Poprosić o liczbę z przedziału od 1 do 5
14. Jeżeli chcemy wstawiać liczby w różne miejsca zbioru, powinniśmy skorzystać z:
   a. tablicy
   b. typu wyliczeniowego
   c. **listy**
15. Odczytując plik używamy obiektu:
   a. System.in
   b. OutputStream
   c. **InputStream**
16. Aby stworzyć własny wyjątek należy dziedziczyć po klasie:
   a. **Exception**
   b. Error
   c. StackTrace
17. Aby umożliwić użytkownikowi wybór kilku opcji w interfejsie graficznym, skorzystamy z:
   a. JComboBox

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

b. **JCheckBox**
c. JRadioButton

18. Aby stworzyć obiekt reagujący na zdarzenia myszy, należy zaimplementować interfejs:
    a. LayoutManager
    b. MouseEvent
    c. **MouseListener**
19. Zarządca rozkładu w interfejsie graficznym to w przypadku Swinga:
    a. **Layout**
    b. Listener
    c. JPanel
20. Plik wykonywalny JAR to:
    a. **Archiwum ZIP**
    b. Plik z pojedynczym kodem bajtowym
    c. Kod maszynowy

8 Lekcje

8.1 Lekcja 1 - Wstęp do Javy i środowiska pracy

8.1.1 Cel lekcji

Celem lekcji jest poznanie czym jest jżyk programowania Java, w jaki sposób działa i jakie sa jego zalety. Wytłumaczone jest jak poprawnie skonfigurować środowisko pracy i co jest potrzebne, aby móc uruchamiać zbudowane programy.

8.1.2 Treść - slajdy z opisem

Slajd 1

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 2

Java jest językiem kompilowanym do kodu bajtowego. Oznacza to, że aplikacje napisane w Javie i skompilowane nie mogą zostać uruchomione bezpośrednio przez system operacyjny. Niezbędna jest maszyna wirtualna.
Java została stworzona przez grupę roboczą pod kierunkiem Jamesa Goslinga z firmy Sun Microsystems, obecnym właścicielem tej technologii jest Oracle Corporation.
Jego podstawowe koncepcje zostały przejęte z języka Smalltalk (maszyna wirtualna, zarządzanie pamięcią) oraz z języka C++ (duża część składni i słów kluczowych).

Główne koncepcje

Autorzy języka Java określili kilkanaście kluczowych koncepcji swojego języka. Najważniejsze z nich to:

Obiektowość

W przeciwieństwie do proceduralno-obiektowego języka C++, Java jest silnie ukierunkowana na obiektowość. O obiekcie można myśleć jako o samoistnej części programu, która może przyjmować określone stany i ma określone zachowania, które mogą zmieniać te stany bądź przesyłać dane do innych obiektów. Wyjątkiem od całkowitej obiektowości są typy proste (inaczej: typy wbudowane, prymitywy).

Dziedziczenie
W Javie wszystkie obiekty są pochodną obiektu nadrzędnego (jego klasa nazywa się po prostu Object), z którego dziedziczą podstawowe zachowania i właściwości. Dzięki temu wszystkie mają wspólny podzbiór podstawowych możliwości, takich jak ich: identyfikacja, porównywanie, kopiowanie, niszczenie czy wsparcie dla programowania współbieżnego.

Niezależność od architektury
Tę właściwość Java ma dzięki temu, że kod źródłowy programów pisanych w Javie kompiluje się do kodu pośredniego (kodu

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

bajtowego). Powstały kod jest niezależny od systemu operacyjnego i procesora, a wykonuje go tzw. wirtualna maszyna Javy, która (między innymi) tłumaczy kod uniwersalny na kod dostosowany do specyfiki konkretnego systemu operacyjnego i procesora. W tej chwili wirtualna maszyna Javy jest już dostępna dla większości systemów operacyjnych i procesorów.Jednak z uwagi na to, że kod pośredni jest interpretowany, taki program jest wolniejszy niż kompilowany do kodu maszynowego.

Sieciowość i obsługa programowania rozproszonego
Dzięki wykorzystaniu reguł obiektowości, Java nie widzi różnicy między danymi płynącymi z pliku lokalnego a danymi z pliku dostępnego przez HTTP czy FTP.

Niezawodność i bezpieczeństwo
W zamierzeniu Java miała zastąpić C++ – obiektowego następcę języka C. Jej projektanci zaczęli od rozpoznania cech języka C++, które są przyczyną największej liczby błędów programistycznych, by stworzyć język prosty w użyciu, bezpieczny i niezawodny. O ile po siedmiu odsłonach Javy jej prostota jest dyskusyjna, o tyle język faktycznie robi dużo, by utrudnić programiście popełnienie błędu.

Slajd 3

Aby móc uruchomić aplikację napisaną w Javie musimy mieć maszynę wirtualną. Znajduje się ona w pakiecie Java JRE (Java Runtime Environment). Można go łatwo wyszukać korzystając z wyszukiwarki Google lub pobrać bezpośredno ze strony www.oracle.com.

Aby móc uruchomić aplikację napisaną w Javie musimy mieć większy zestaw narzędzi o nazwie Java JDK (Java Development Kit). Znajduje się tam przede wszystkim kompilator. Cały pakiet dostępny jest również na stronie www.oracle.com.

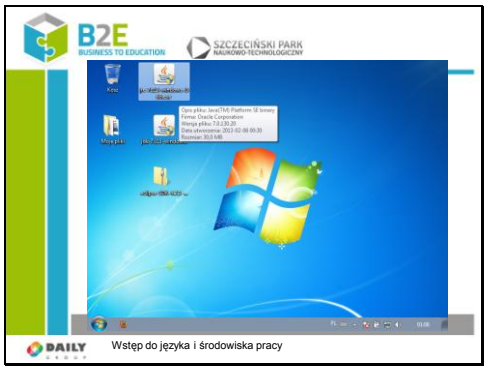
Należy przede wszystkim pamiętać, że Java stworzona jest na licencji GNU

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

(General Public License). Na slajdzie przedstawione są jedynie oficjalne narzędzia. Każdy może napisać własną maszynę wirtualną i istnieje wiele alternatywnych źródeł. Np. otwarta implementacja Javy - OpenJDK.

Bardzo ważnym elementem jest również środowisko pracy. Stanowczo odradzam korzystanie z prostych edytorów tekstowych. Jednym z najpopularniejszych środowisk jest Eclipse. Można pobrać również środowisko NetBeans lub IntelliJ IDEA. Wszystkie trzy oferują podobne możliwości i są powszechnie wykorzystywane na całym świecie. Zalecam pobranie pakietu Eclipse Classic, ponieważ właśnie w nim będę pokazywał przykłady.

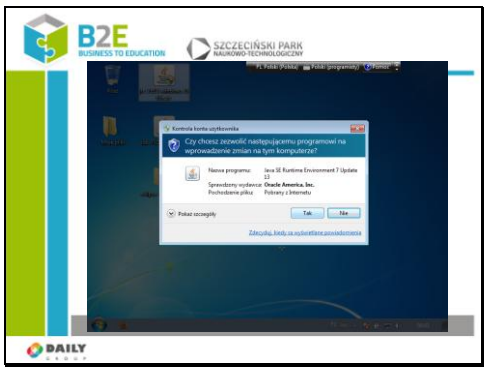
Slajd 4



W pierwszej kolejności instalujemy pakiet Java JRE

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 5



Slajd 6

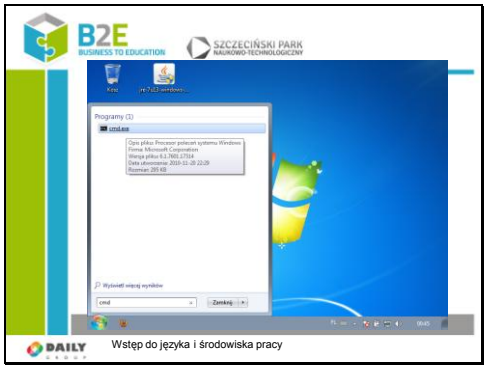
Przechodzimy kolejne kroki instalacji.

Slajd 7

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 8

Slajd 9

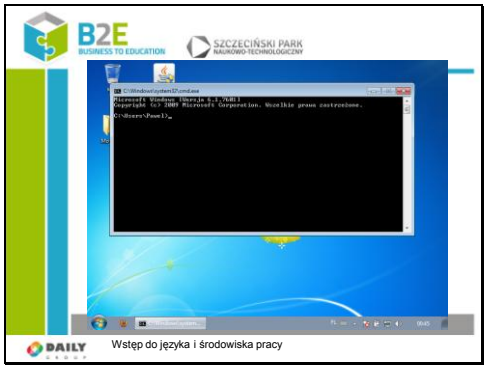


Po zakończeniu instalacji możemy korzystać z oprogramowania stworzonego w języku Java.

W dalszych slajdach pokazane będzie jak uruchomić aplikację. Dla zaprezentowania posiadam już gotowy plik wykonywalny. Proszę nie wykonywać tej części samodzielnie.

W tym celu należy uruchomić program cmd.exe.

Slajd 10



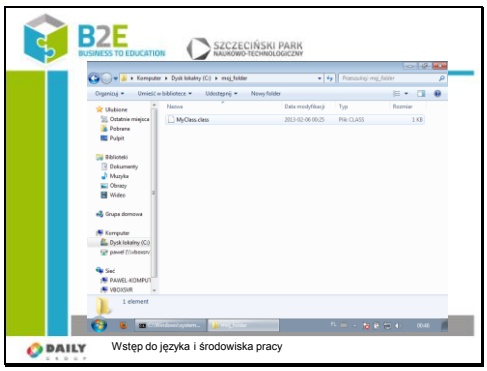
Człowiek - najlepsza inwestycja

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

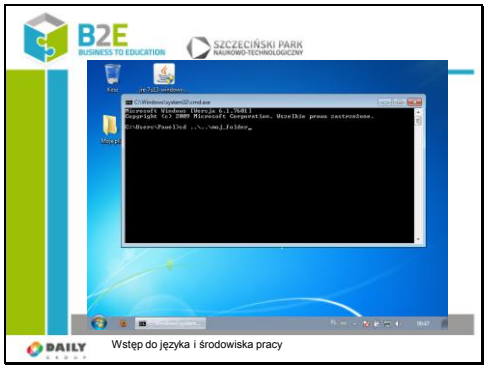
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 11



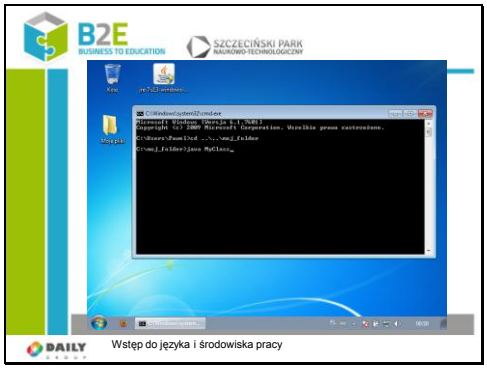
Pliki wykonywalne dla maszyny wirtualnej Javy posiadają rozszerzenie class.

Slajd 12



Przechodzimy do folderu w którym znajduje się nasza klasa.

Slajd 13



Poleceniem java uruchamiamy maszynę podając jako parametr nazwę pliku bez rozszerzenia.

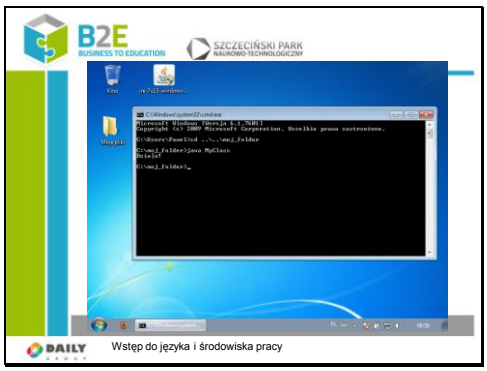
Człowiek - najlepsza inwestycja

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 14

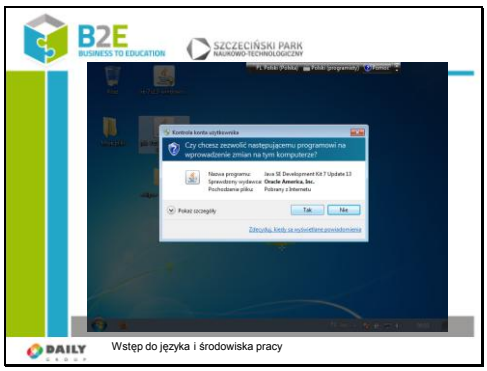


W wyniku otrzymaliśmy napis „Działa!”. Nie wyświetliła się jednak literka „ł”. Spowodowane jest to kodowaniem znaków cp1250 w programie cmd.exe. W Javie symbole zajmują 16 bitów, więc bez żadnych ograniczeń można stosować znaki regionalne.

Slajd 15

Aby móc zacząć programować w Javie potrzebujemy kompilatora z pakietu Java JDK.

Slajd 16



Kolejno wykonujemy polecenia instalatora.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

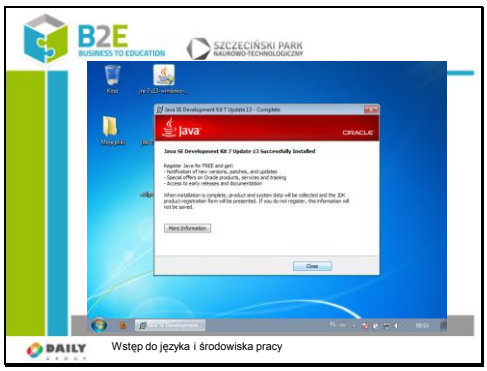
Slajd 17

Slajd 18

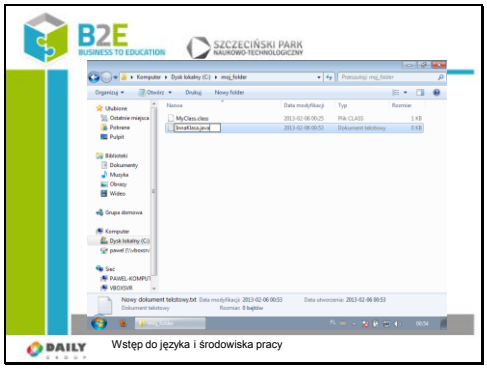
Slajd 19

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 20



Slajd 21

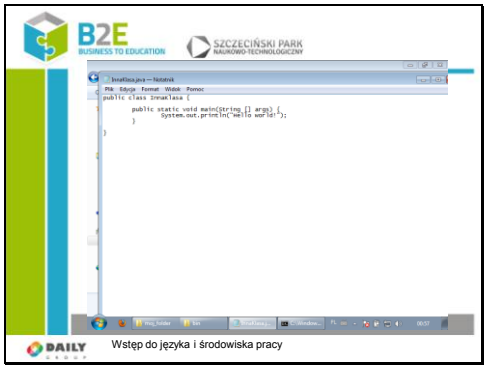


Po zakończonej instalacji Java JDK. Możemy napisać pierwszy program.

Możemy już wykonać pierwsze praktyczne ćwiczenie.

Stwórzmy plik o dowolnej nazwie, bez białych znaków i zaczynający się dużą literą, z rozszerzeniem *.java.

Slajd 22

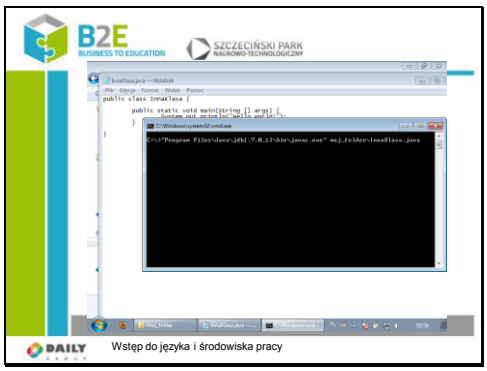


Do pliku wstawiamy ten fragment kodu.

Uwaga!
Jeżeli użyta została inna nazwa pliku niż „InnaKlasa” to należy ją wpisać w pierwszym wersie (bez rozszerzenia) zamiast wyrażenia „InnaKlasa”.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 23

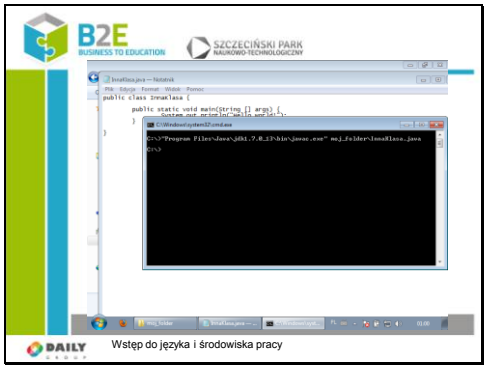


Kompilujemy nasz kod źródłowy widocznym poleceniem.

Pierwsza część jest ścieżką do kompilatora Javy – javac.exe.

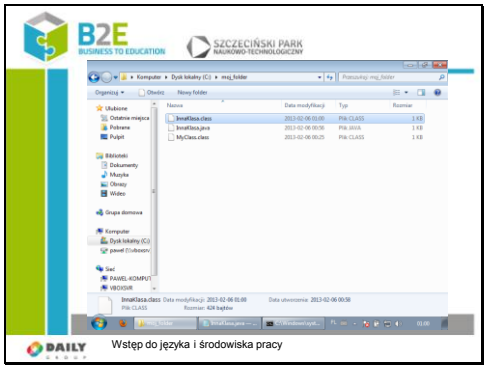
Argumentem przyjmowanym przez kompilator jest ścieżka do pliku z kodem źródłowym.

Slajd 24



Tak wygląda rezultat poprawnie wykonanego polecenia kompilacji (brak jakichkolwiek wiadomości i ponowne wyświetlenie znaku zachęty).

Slajd 25

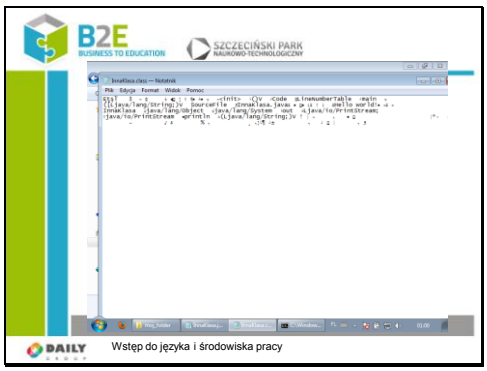


Jedyną widoczną zmianą jest pojawienie się pliku o takiej samej nazwie jaką wybraliśmy do pliku z rozszerzeniem „*.java”.

Nowy plik posiada rozszerzenie „*.class”. Zawiera on kod bajtowy przeznaczony do uruchamiania na każdej maszynie wirtualnej.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

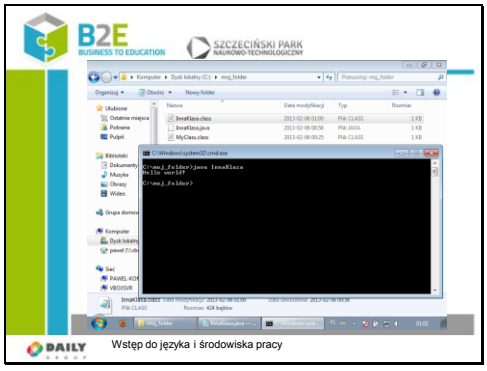
Slajd 26



Możemy wyświetlić zawartość nowego pliku w edytorze tekstowym.

Jak widać nie jest to kod maszynowy. Wiele części można odczytać, np. „SourceFile" jest to plik źródłowy z którego powstał dany kod bajtow, „java/lang/String" jest to adres obiektu, który przechowuje ciąg znaków.

Slajd 27



Omawiany plik możemy uruchomić poleceniem pokazanym w poprzednim przykładzie.

Slajd 28

Pisanie kodu w prostym edytorze tekstowym jest trudne. Będziemy korzystać ze środowiska Eclipse w celu usprawnienia pracy.

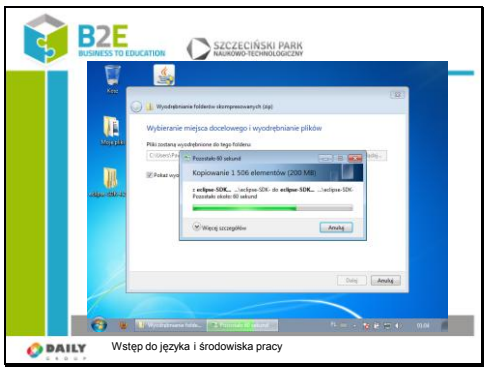
Człowiek - najlepsza inwestycja

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

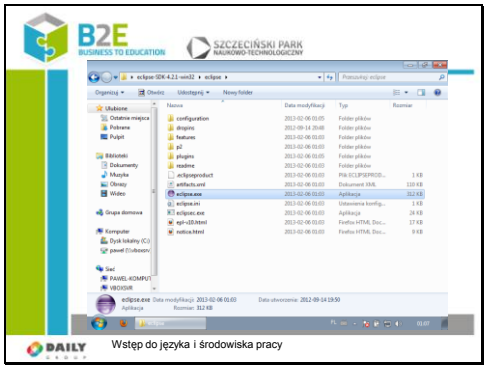
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 29



Wypakowujemy pobrane archiwum.

Slajd 30



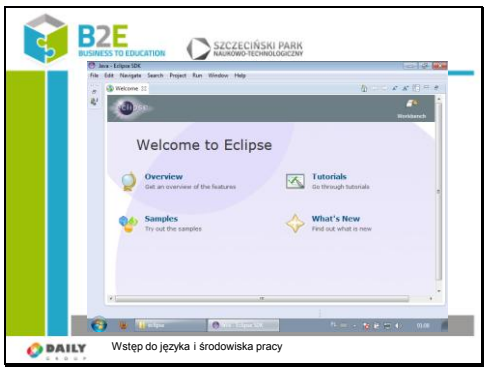
Środowisko uruchamiamy za pomocą pliku „eclipse.exe”.

Slajd 31

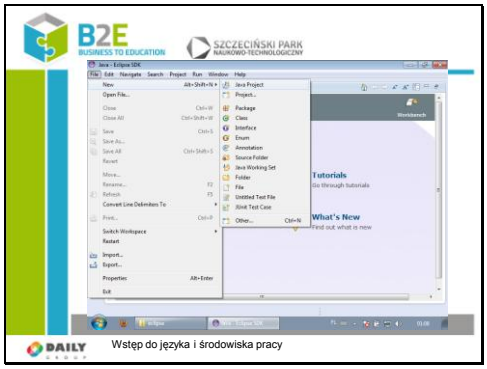
Przy pierwszym uruchomieniu trzeba wybrać folder pełniący funkcję „workspace”. Jest to lokalizacja w której znajdować się będą wszystkie nasze projekty.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 32



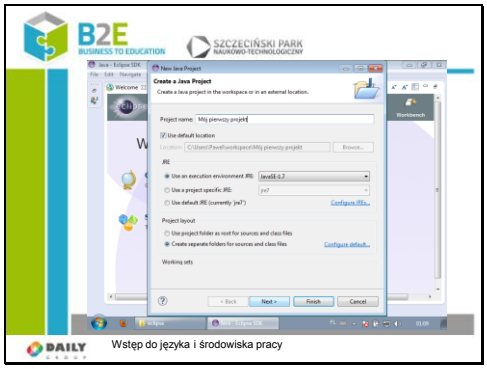
Slajd 33



Zanim przystąpimy do pisania kodu, musimy stworzyć projekt.

W tym celu wybieramy:
File/New/Java Project

Slajd 34



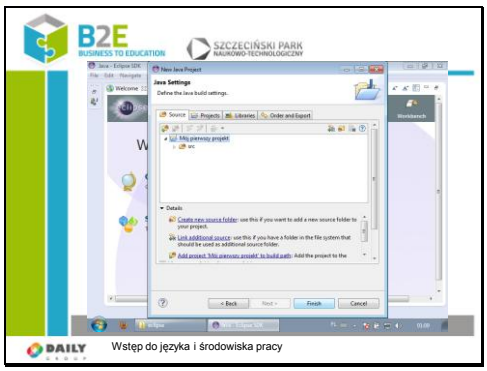
Nazwa projektu (Project name) może być dowolna.

Pozostałe parametry najlepiej pozostawić z ustawieniami domyślnymi.

Klikamy „Next”.

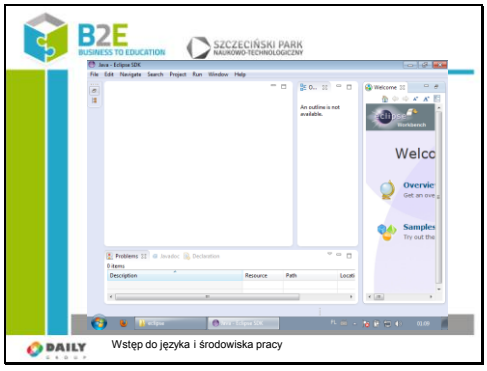
Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 35



Klikamy „Finish”.

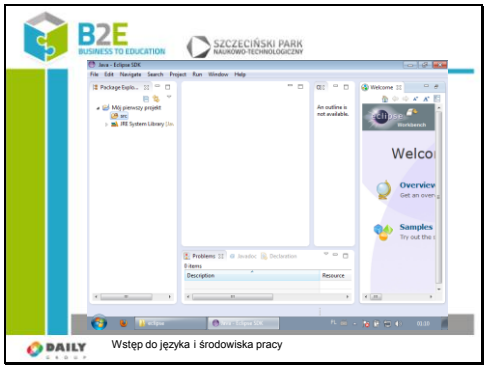
Slajd 36



Aby zobaczyć zawartość projektu klikamy na ikonę znajdującą się w lewym, górnym rogu (poniżej pozycji w menu o nazwie „File”).

Powinien wyświetlić się nam „Package Explorer”

Slajd 37



Pliki kodu źródłowego przechowywane są w folderze „src”.

Nowy projekt nie posiada żadnych plików. W celu stworzenia nowego należy kliknąć na folderze „scr” prawym przyciskiem myszy.

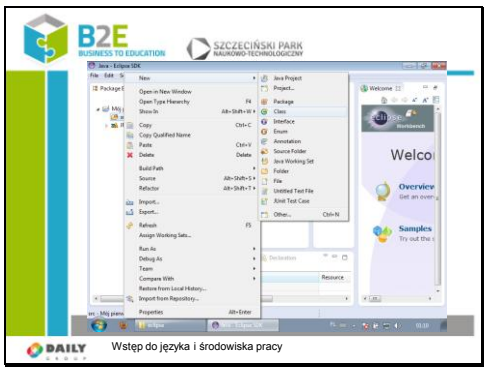
Człowiek - najlepsza inwestycja

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

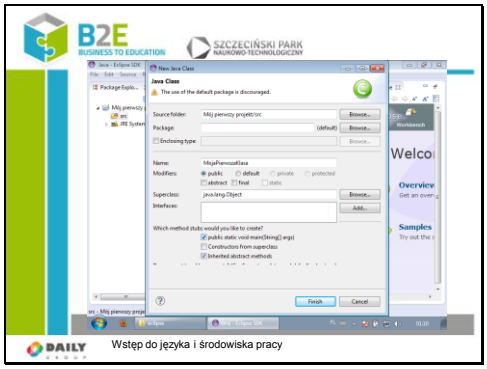
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 38



Następnie wybieramy: New/Class.

Slajd 39

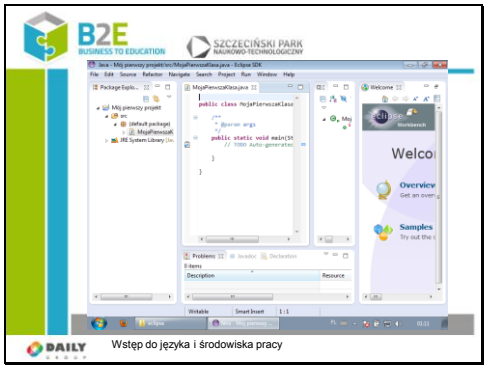


Podajmy nazwę klasy. Nie używamy spacji i zaczynamy dużą literą.

Zaznaczamy również opcję: „public static void main (String [] args)”.

Całość zatwierdzamy przyciskiem „Finish”.

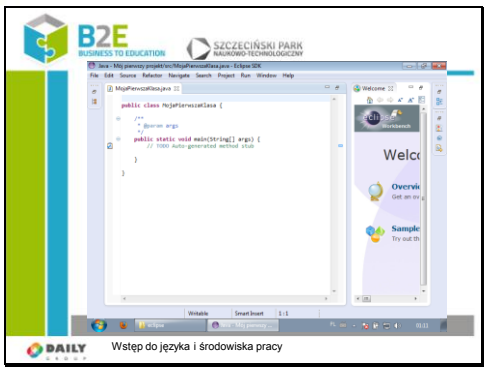
Slajd 40



Na środku ekranu wyświetliła się zawartość pliku. Znajduje się on w folderze „src” w (defaultpackage)”.

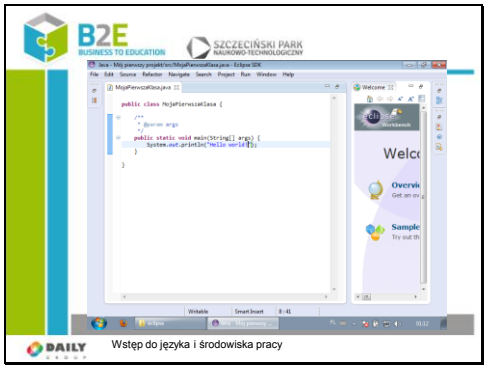
Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 41



Jeżeli chcemy wyświetlić plik na całej szerokości ekranu, klikamy podwójnie na zakładce z zawartością pliku.

Slajd 42

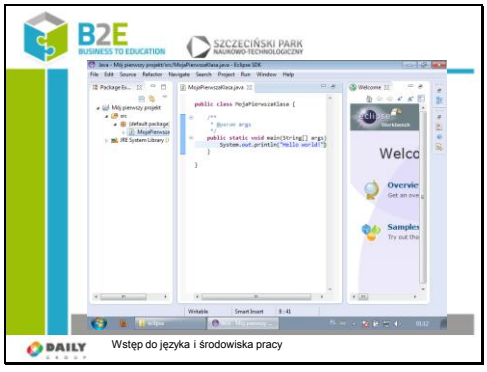


Wers zaczynający się na „//” zastępujemy wyrażeniem:

System.out.println("Hello world!");

Skrótem klawiszowym „Ctrl” + „S” zapisujemy zmiany.

Slajd 43



Ponownie klikamy podwójnie na nazwie karty w celu jej zmniejszenia.

Dzięki temu ponownie widzimy strukturę plików w projekcie.

W celu uruchomienia pliku klikamy w obszarze „Package Explorer” na nim prawym klawiszem myszy.

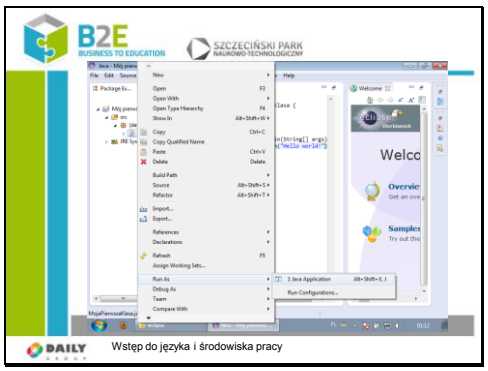
Człowiek - najlepsza inwestycja

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

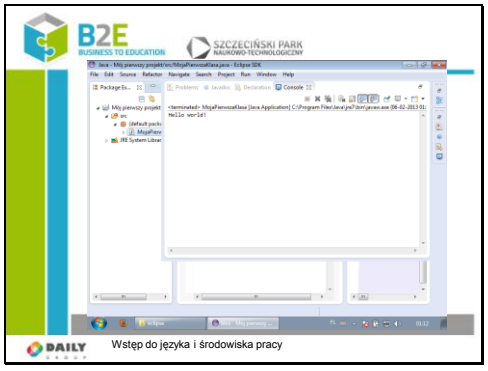
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 44



Wybieramy opcję:
„Run As/1 Java Aplication”

Slajd 45



Jako rezultat operacji powinniśmy zobaczyć konsolę z widocznym napisem „Hello world!”.

Slajd 46

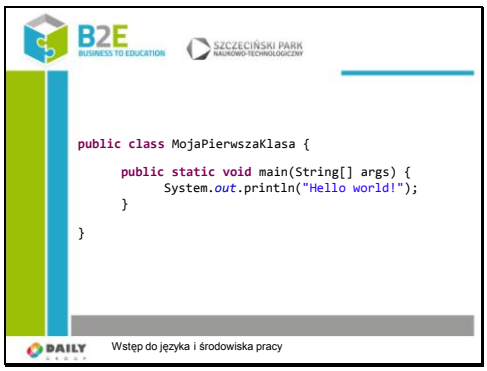
Człowiek - najlepsza inwestycja

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 47



Przeanalizujmy kod pierwszego programu.

Wszystko w Javie jest obiektem. Obiekt jest instancją danej klasy, czyli pewnym tworem przechowywanym w pamięci, który przechowuje dane i nie tylko. Pojęcie obiektu zostanie dokładnie wytłumaczone na lekcji 3.

W Javie w każdym pliku znajduje się jedna klasa. Później przedstawię jak definiować klasy wewnątrz innych klas. Jednak po procesie kompilacji zawsze klasa jest osobnym plikiem.

Bardzo ważne jest aby klasa miała taką samą nazwę jak plik, w którym się znajduje. Pominięcie tej zasady traktowane jest jako błąd!

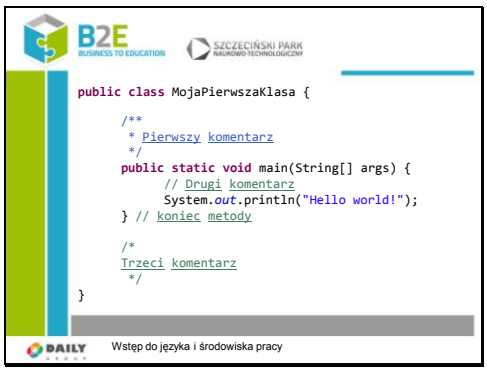
Wewnątrz klasy (pomiędzy nawiasami „{” i „}”) znajdują się pola i metody. W naszym przykładzie nie ma pól, jest tylko jedna metoda o nazwie „main”.

W późniejszych lekcjach zostaną również wytłumaczone znaczenia słów „public”, „static” i „void”. Na razie wszystkie polecenia będziemy pisać we wnętrzu metody „main”. Jest to szczególna metoda, ponieważ jest zawsze wywoływana przy próbie uruchomienia obiektu.

Obiekt przeznaczony do, bezpośredniego uruchomienia musi posiadać metodę:
„pubic static void main(String [] args)”.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 48



W Javie, w prawie dowolnym miejscu, możemy pisać komentarze. Są one przeznaczone wyłącznie dla programistów.

Najczęściej spotykanymi komentarzami są te zaczynające się od symboli „//”. Możemy stosować je w tych samych liniach, w których są elementy kodu. Po wstawieniu symboli „//” kompilator pominie cały tekst, aż do końca linii.

Innym typem komentarzy są komentarze blokowe. Otwieramy je symbolem „/*” i zamykamy „*/”. Pomiędzy tymi znacznikami możemy wstawić dowolną liczbę wersów.

Dość szczególnym typem komentarzy blokowych są te zaczynające się symbolami „/**”. Zostawia się je przed definicją klasy i jej metodami. Nazywane są „Javadoc”. Pełnią ważną funkcję w tworzeniu dokumentacji. W późniejszych lekcjach zostanie pokazane jak je tworzyć, oraz jak korzystać z tej dokumentacji.

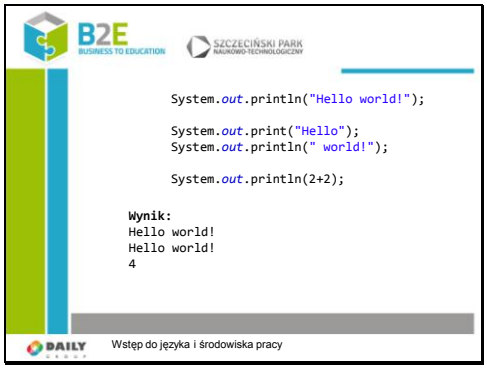
Pisanie komentarzy jest bardzo ważne. W dużych projektach, nad którymi pracuje wiele osób, są one praktyczną formą zostawiania wiadomości. Jeżeli napiszemy fragment kodu, o bardzo skomplikowanej logice, warto go skomentować. W takim przypadku bardzo przydatne mogą okazać się nawet lakoniczne informacje o podstawowych zasadach działania kodu. Bardzo ułatwią one rozwijanie aplikacji osobą, które będą pracowały po nas.

Praca zespołowa nie jest jedynym powodem, dla którego pisze się komentarze. Jeżeli zechcemy po pewnym czasie powrócić do rozwijania swojej aplikacji, możemy nie pamiętać szczegółów dotyczących kodu. Oszczędzimy wiele czasu czytając komentarze, zamiast szczegółowo analizować kod.

Dla kompilatora bieżący kod jest identyczny jak ten na poprzednim slajdzie.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 49



Metoda:
„System.out.println”
Wyświetla tekst podany w nawiasach (i w znakach „”) w konsoli i wstawia znak nowej linii.

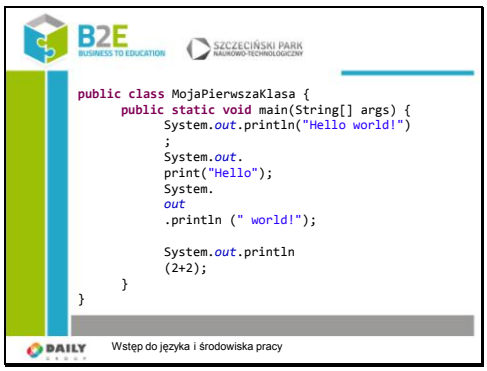
W pakiecie „System.out” są również inne metody, np. „print” wstawia tekst bez znaku nowej linii.

Metoda „println” wyświetla nie tylko tekst. W Javie wszystkie obiekty mogą być wyświetlone w postaci tekstu. Będziemy wykorzystywać ją na najbliższych lekcjach do sprawdzania wyników przebiegu programów.

Jest to jedna z najpowszechniej używanych metod. W środowisku Eclipse możemy wywołać ją automatycznie wpisując „sysou” i następnie wciskając „Ctrl” + „spacja”.

Skrót klawiszowy „Ctrl” + „spacja” jest powszechnie wykorzystywany (nie tylko w Eclipse!) do podpowiadania elementów składni. Będziemy często go wykorzystywać.

Slajd 50



W Javie każde polecenie musi kończyć się średnikiem. Nie oznacza to jednak, że średnik musi znajdować się na końcu każdej linii. Jeżeli nasze polecenia są zbyt długie (nie zaleca się stosować wersów o długości 200 znaków, są nieczytelne), możemy umieścić je w kilku wersach, pamiętając o zakończeniu całości średnikiem

Na slajdzie znajduje się poprawny kod programu, wykonujący to co zostało przedstawione na slajdzie poprzednim. Należy zauważyć jednak, że polecenia zapisane w taki sposób nie są czytelne dla ludzi.

Średnika nie wstawiamy, np. po nawiasach klamrowych.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

8.1.3 Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie poznają teoretyczna widzę na temat tego czym jest Java. Będą umieli rozpocząć pracę w Javie w odpowiednim środowisku.

8.2 Lekcja 2 - Typy podstawowe i instrukcje sterujące

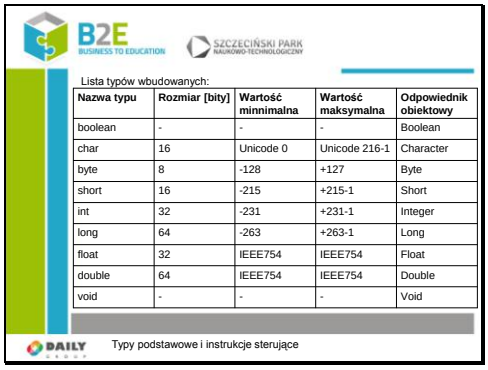
8.2.1 Cel lekcji

Celem lekcji jest poznanie podstawowych typów danych, wytłumaczenie jak bezpiecznie przechowywać dane w zmiennych. Poznanie pętli sterujących i zarządzanie przebiegiem wykonania programu za ich pomocą.

8.2.2 Treść - slajdy z opisem

Slajd 1

Slajd 2



Nazwa typu	Rozmiar [bity]	Wartość minnimalna	Wartość maksymalna	Odpowiednik obiektowy
boolean	-	-	-	Boolean
char	16	Unicode 0	Unicode 216-1	Character
byte	8	-128	+127	Byte
short	16	-215	+215-1	Short
int	32	-231	+231-1	Integer
long	64	-263	+263-1	Long
float	32	IEEE754	IEEE754	Float
double	64	IEEE754	IEEE754	Double
void	-	-	-	Void

Typy wbudowane są szczególnym typem danych. Występują ponieważ traktowanie prostych danych jako obiekty nie jest szczególnie wydajne. Zmienne te przechowują proste wartości

Typ „boolean” służy do przechowywania wartości logicznej: prawda lub fałsz.

Typ „char” służy do przechowywania znaku (liter). Posiada 16 bitów, więc możemy swobodnie używać polskie znaki.

Typ „byte” służy do przechowywania bajtów danych.

Typy „short”, „int” i „long” służą do przechowywania liczb całkowitych. W zależności od zakresu na jakim chcemy operować wybieramy odpowiedni typ.

Człowiek - najlepsza inwestycja

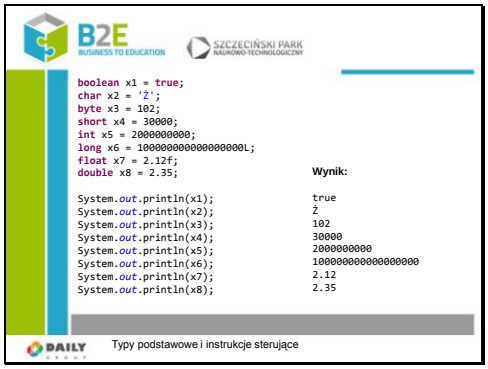
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI · UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Typy „float” i „double” przechowują liczby zmiennoprzecinkowe zgodnie ze standardem IEEE754.

Typ „void” oznacza zerową informację. Nie można stworzyć zmiennej typu „void”, ale może on być zwracany przez metody.

Slajd 3



Na slajdzie pokazane jest jak powinno się inicjalizować zmienne (nadawać wartości):

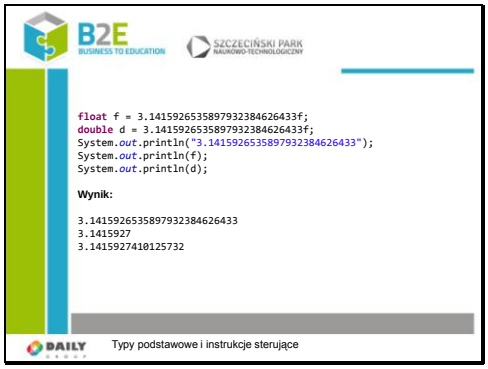
- boolean – używamy słów „true” lub „false”,
- char – podajemy pojedynczy znak w apostrofach,
- byte, short, int – liczba całkowita,
- long – liczba całkowita z literą „l” lub „L” na końcu,
- float – liczba ułamkowa z literą „f” lub „F” na końcu,
- double – liczba ułamkowa.

W celu wyświetlenia zainicjowanych wartości używamy wcześniej przedstawionej metody „println”.

Jak widać w wyniku nasze zmienne przechowują dokładnie takie wartości jak podaliśmy.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 4

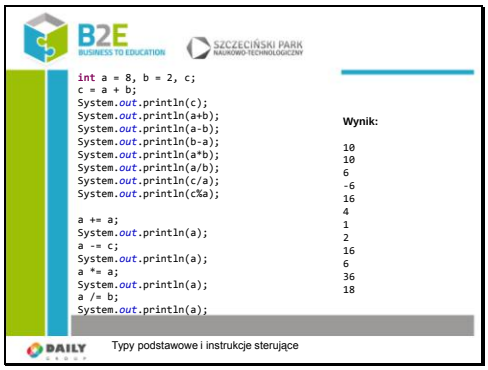


Dokonując operacji na typach zmiennoprzecinkowych należy szczególnie pamiętać o dokładności wykonywanych obliczeń.

Przykład pokazuje dokładność z jaką typy „float" i „double" przechowują liczbę Pi.

Nie oznacza to jednak, że w Javie nie można przeprowadzać obliczeń o wyższej precyzji. W standardowych bibliotekach Javy znajdują się klasy takie jak „BigInteger" i „Bigdecimal", które mogą wykorzystać do zapamiętania liczby całą dostępną pamięć RAM komputera. Są to jednak obiekty. Jak korzystać z obiektów zostanie wytłumaczone w następnej lekcji.

Slajd 5



Na typach wbudowanych możemy wykonywać operacje matematyczne korzystając z dostępnych operatorów.

Operator „%" służy do otrzymywania reszty z dzielenia, pozostałe działają zgodnie z intuicją.

Przypisanie sumy liczb a i b do c nastąpiło poprzez użycie operatora podstawiania („=").

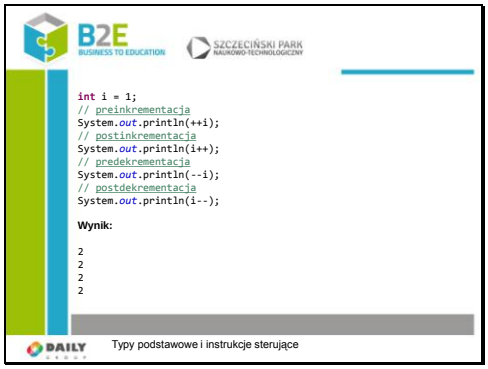
Operatory „+", „-" i „*" działają zgodnie z intuicją (suma, różnica, mnożenie).

Należy zwrócić szczególną uwagę na operator „/" dzielenia. W przypadku gdy dzielimy a przez b dostajemy poprawny wynik (8/2=4). Kiedy jednak dzielimy c przez a (10/8=1,25) otrzymaliśmy wynik 1 zamiast 1,25. Dzieje się tak, ponieważ typy te nie przechowują części ułamkowej. Nie następuje również zaokrąglenie wartości. Cyfry znajdujące się po przecinku są „odcinane", więc zamiast np. 1,8 też dostaniemy 1.

Operatory użyte poniżej biorą argumenty z obu stron, przypisując wartość do zmiennej znajdującej się z lewej strony. Wykonują analogicznie operacje sumy, różnicy, mnożenia i dzielenia.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 6



Operator preinkrementacji najpierw zwiększa zmienną o jedną jednostkę, następnie zwraca jej wartość. 1 zostało zwiększone, a następnie wyświetlone. Po operacji i=2.

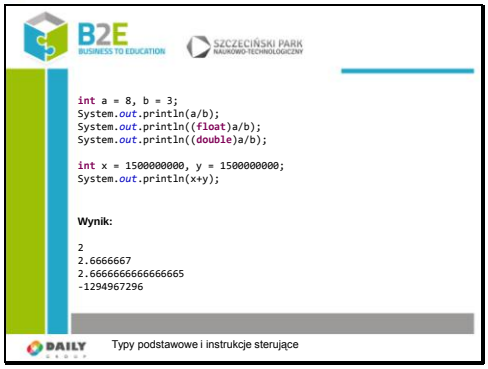
Operacja postinkrementacji najpierw zwraca wartość, a następnie ją zwiększa 2 zostało najpierw wyświetlone, a następnie zwiększone. Po tej operacji i=3.

Operator predekrementacji najpierw zmniejsza zmienną o jednostkę, następnie zwraca jej wartość. 3 zostało zmniejszone, a następnie wyświetlone. Po operacji i=2.

Operacja postdekrementacji najpierw zwraca wartość, a następnie ją zmniejsza. 2 zostało najpierw wyświetlone, a następnie zmniejszone. Po tej operacji i=1.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 7



Fakt, że dla typów „int”, „short”, „long” wartości mniejsze od 1, są obcinane, nie oznacza, że nie możemy ich używać jeżeli w przyszłości może zajść potrzeba dokonania dokładniejszych obliczeń. Istnieje możliwość rzutowania jednych typów na drugie.

Na slajdzie mamy trzy możliwe wyniki dzielenia 8 przez 3 w zależności od dokładności wyniku jaki chcemy otrzymać.

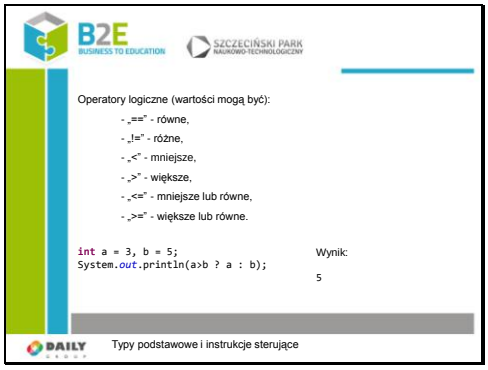
Poniżej mamy dość szczególny przypadek. Zastanówmy się, jak to jest możliwe, że dodając dwie liczby dodatnie, otrzymaliśmy ujemny wynik.

Proszę wyobrazić sobie, że typy dla liczb całkowitych mogą się „przekręcać”.

Dobrą analogią do tego zjawiska jest przykład licznika samochodowego. Jeżeli wartość na nim pokazywana będzie cały czas rosnąć, to przekroczy wartość maksymalną i ponownie przekroczy zero. Z omawianymi typami dziej się dokładnie tak samo.

Wartość maksymalna dla „int” to 231-1 (około 2 mld). Jeżeli zechcemy zapisać 3 mld, to „licznik się przekręci” i do minimalnej wartości -231 (około -2 mld) dodana zostanie wartość będąca różnicą 3 mld – 2mld = 1 mld. Otrzymaliśmy w ten sposób wynik około -1 mld.

Slajd 8



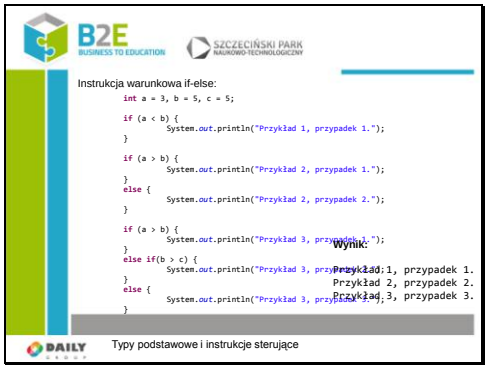
Przedstawione operatory służą do porównywania wartości. Działają zgodnie z tym, czego używaliśmy na lekcjach matematyki.

Poniżej mamy przykład jedynego trójargumentowego operatora operatora w Javie. Składa się z dwóch znaków: „?” i „:”. W pierwszej kolejności podajemy wartość logiczną. Wynikiem każdej operacji logicznej jest wartość logiczna. Możemy w tym miejscu równie dobrze użyć zmiennej typu boolean. Następnie, po znaku „?” podajemy polecenie, które ma zostać wykonane w przypadku prawdy. Po znaku „:” podajemy polecenie dla fałszywej wartości logicznej.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Wynikiem użycia operatora „?” w przykładzie pokazanym na slajdzie jest drukowanie w konsoli wartości większej (a lub b).

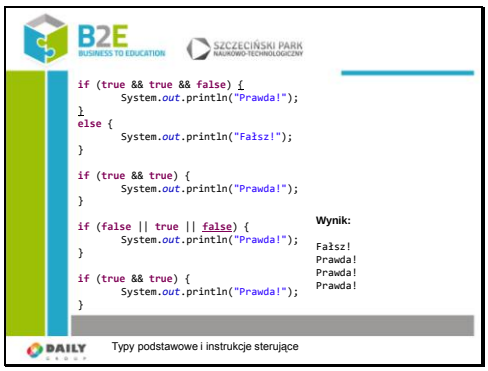
Slajd 9



Na slajdzie przedstawione zostały różne warianty instrukcji warunkowej „if”. Korzystając z niej decydujemy, które polecenia mają być wykonane w zależności od zadanych kryteriów.

Instrukcja warunkowa „if” działa w następujący sposób. Warunek w nawiasie po słowie „if” musi być wyrażeniem logicznym (dawać wartość prawda lub fałsz). W przypadku prawdy wykonają się instrukcje zawarte w nawiasach klamrowych, za wyrażeniem logicznym. Jeżeli później występuję słowo „else”, to w przypadku prawdy wykonają się instrukcje występujące po słowie else.

Slajd 10

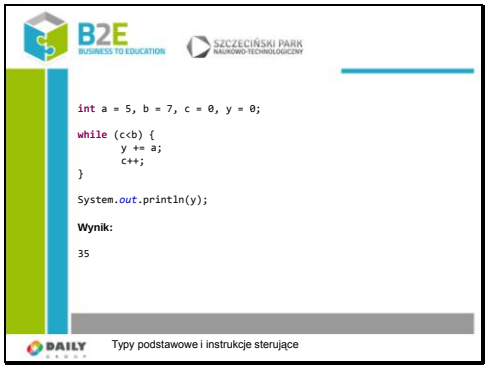


Dostępne mamy również inne operatory logiczne: koniunkcji („&&”) i alternatywy(„||”).

Należy pamiętać, że kiedy użyjemy operatora „&&”, to kiedy przynajmniej jedna z wartości jest fałszywa, całe wyrażenie jest fałszywe. Dlatego wyrażenia sprawdzane są od lewej do prawej. Po napotkaniu pierwszej fałszywej wartości reszta nie jest już sprawdzana. Nie zachodzi taka potrzeba.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 11

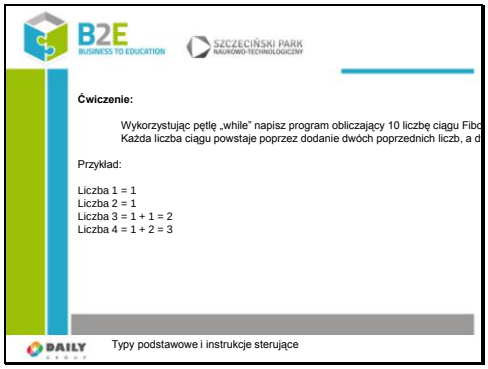


Na slajdzie widzimy przykładową pętlę „while". Wykonuje ona operację mnożenia a i b, zapisując wynik do c.

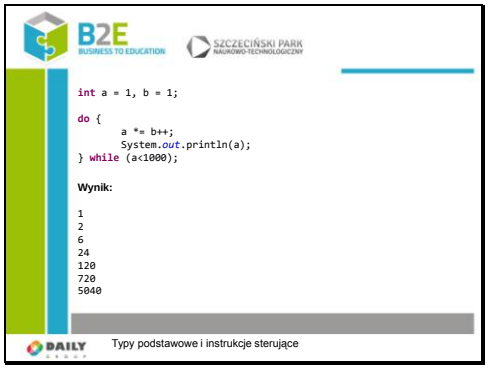
Po słowie „while" występuje wyrażenie logiczne, tak jak w instrukcji „if". W przypadku prawdy instrukcje z nawiasów klamrowych zostaną wykonane. Następnie wyrażenie logiczne jest sprawdzane ponownie i w przypadku prawdy następuje kolejny krok pętli. Instrukcja „while" przestaje działać jeżeli warunek (wyrażenie logiczne w nawiasach) jest fałszywy.

Jeżeli źle skonstruujemy warunek logiczny, pętla może się nie zatrzymać! W takim przypadku proces w systemie operacyjnym, który odpowiada naszemu programowi, powinien być „zabity".

Slajd 12



Slajd 13

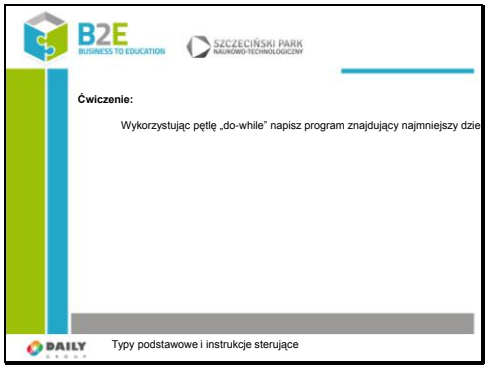


Pętla „do-while" działa tak samo jak pętla „while", z tą różnicą, ze zawsze występuje pierwszy przebieg pętli Dopiero po nim sprawdzany jest warunek. W przypadku prawdy następuje kolejny przebieg.

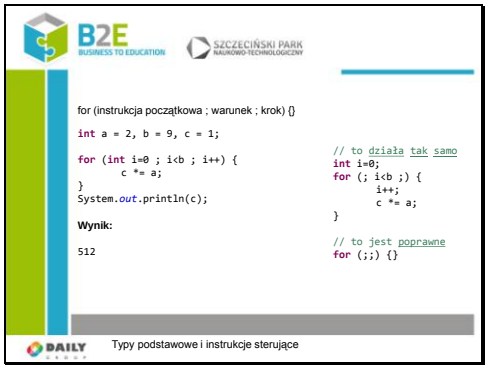
Pętla widoczna na slajdzie wypisuje kolejne wartości silni, aż do momentu, kiedy wartość będzie większa niż 1000.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 14



Slajd 15



Pętla „for" jest bardziej złożona od poprzednich przykładów. W nawiasie po słowie „for" nie występuje tylko warunek. Wyrażanie ma 3 części:

Pierwsza część, instrukcja początkowa, wykonywana tylko raz na samym początku. W naszym przypadku zadeklarowaliśmy i zainicjowaliśmy zmienną i.

Druga część, warunek, jest sprawdzana w każdym kroku. Musi to być wyrażenie logiczne. Dopóki wartość jest prawdziwa, dopóty wykonuje się pętla.

Trzecia część, to instrukcja, która wykonuje się po zakończeniu każdego kroku.

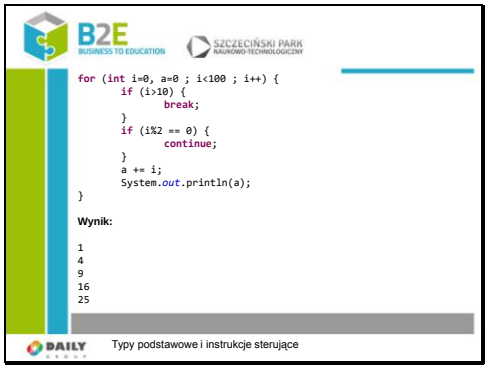
Przedstawiony przykład oblicza wartość liczby a podniesioną do potęgi b.

Każda z trzech instrukcji pętli „for" może być pusta. Z prawej strony slajdu pokazany jest przykład, który działa identycznie jak omawiana pętla. Jednak takie zapisywanie pętli jest mało czytelne dla ludzi.

Pętla „for", przedstawiona w prawym dolnym rogu, również jest poprawna. Powoduje ona zawieszenie się wykonywanego programu, z powodu braku instrukcji warunkowej. Jest to przykład pętli nieskończonej.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 16



Przy operowaniu na pętlach możemy skorzystać z dwóch instrukcji: „break” i „continue”.

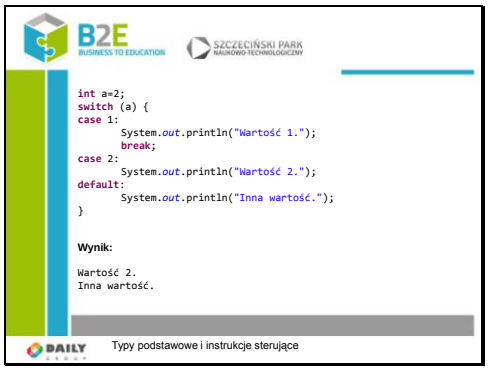
Przy napotkaniu słowa „continue”, wykonanie bieżącego kroku pętli zostaje zatrzymane i zaczyna się wykonywanie kroku następnego.

Po wystąpieniu słowa „break” następuje całkowite wyjście z pętli „for”.

Powyższy przykład sumuje wszystkie liczby nieparzyste od 1 do 10. Instrukcja warunkowa w pętli „for” w naszym przypadku nie ma żadnego znaczenie. Polecenie „break” przerywa działanie pętli kiedy tylko i będzie większe od 10.

Kiedy reszta z dzielenia i przez 2 będzie równa 0 (liczba parzysta), nie nastąpi operacja sumy i wypisania na ekran.

Slajd 17



Ostatnią pętlą jest pętla „switch”. Dzięki niej możemy decydować, które instrukcje wykonają się w danym przypadku.

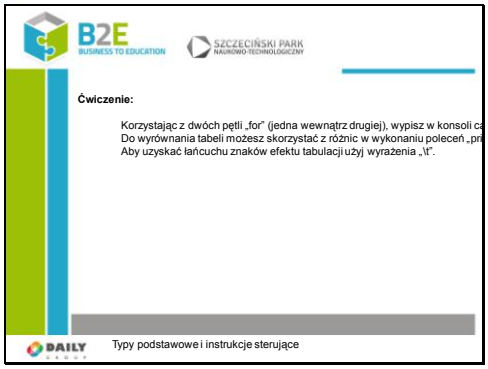
Po słowie „switch” podajmy w nawiasie wartość typu „int”, „String” lub „Enum” (nie może to być wartość logiczna). Pojęcia „String” i „Enum” zostaną wytłumaczone w następnych lekcjach.

Następnie możemy zdefiniować dowolną ilość przypadków. Po słowie „case” podajmy wartość dla której dane instrukcje mają się wykonać. Nie otwieramy nawiasu klamrowego! Pamiętajmy, aby wstawić „break”, jeżeli nie chcemy, aby wykonały się instrukcje dla następnego przypadku. Po drugim „case” nie ma tego słowa, więc wykonały się również instrukcje dla przypadku „default”.

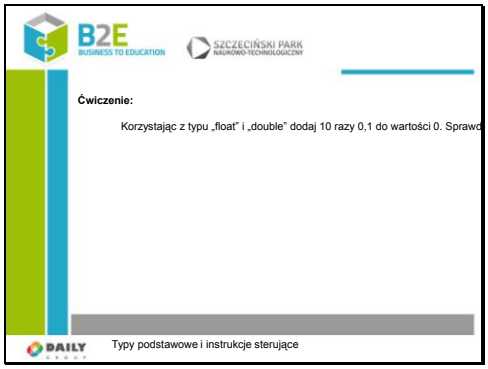
„default” jest przypadkiem szczególnym. Jeżeli wartość a nie występuje w żadnym z „case”, wówczas wykonają się instrukcje z tej części.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

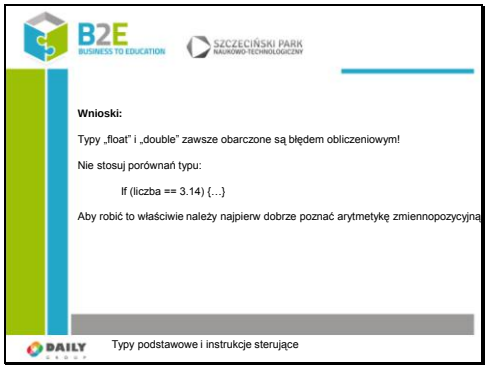
Slajd 18



Slajd 19



Slajd 20



8.2.3 Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie będą wiedzieli jak poprawnie przechowywać dane w zmiennych, oraz jak programować pętle do wykonywanie odpowiednich operacji.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

8.3 Lekcja 3 - Wstęp do programowania obiektowego

8.3.1 Cel lekcji

Celem lekcji jest wytłumaczenie na czym polega idea programowania obiektowego. Opis budowy klasy w Jawie, konstruktorów i metod.

8.3.2 Treść - slajdy z opisem

Slajd 1

Slajd 2

Niemal każdy element świata rzeczywistego możemy potraktować jako obiekt, niezależnie od tego, czy jest niepodzielnym elementem (np. kamień), czy składa się z wielu innych części (np. długopis).

Jest to naturalny sposób postrzegania świata.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 3

O każdy obiekt ma typy właściwości:

Informacje:
- kolor długopisu,
- cena,
- prędkość maksymalna samochodu,
- masa;

Operacje:
- pisanie,
- zdolność do poruszania się,
- wydawanie dźwięków.

Zauważmy, że ciężko jest wyobrazić sobie temperaturę jako obiekt. Nie utożsamiajmy obiektu z przedmiotem! Obiektem może być np. stan techniczny części samochodowej.

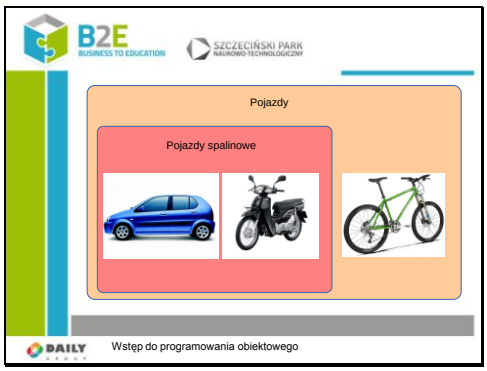
Myślę, że dobrym przykładem na to jest temperatura:
- ma swoja wartość,
- może maleć i wzrastać,
- nie może być niższa od zera absolutnego.

Jeżeli chcielibyśmy stworzyć model pomieszczenia, w którym się znajdujemy, obiektem powinno być przede wszystkim samo pomieszczenie, przedmioty znajdujące się w środku, jak i temperatura powietrza.

Byłby to model obiektowy. W jasny i czytelny sposób pozwoliłby zachować niezbędne informacje na temat stanu wszystkich przedmiotów, oraz przewidzieć możliwe zdarzenia. Jeżeli jest tablica i kreda, to możliwe, że któraś z osób coś na niej napisze. Jeżeli nie znalazłby się obiekt z operacją „pisz” (kreda lub długopis), to z obiektowego punktu widzenia niemożliwe jest, aby w tym pomieszczeniu zostało coś napisane.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 4



Na obiektach możemy dokonywać generalizacji i specjalizacji. Są to kluczowe zjawiska, jeżeli chcemy skorzystać z mechanizmu dziedziczenia. Pojęcie dziedziczenia zostanie wyjaśnione na poprzedniej lekcji.

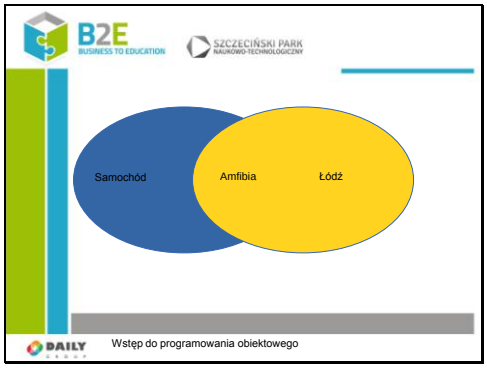
Samochód jest pojazdem spalinowym, jeżeli powiemy, że jest tylko pojazdem, wtedy następuje generalizacja.

Jeżeli powiemy, że samochód jest pojazdem, a ponadto jest pojazdem spalinowym, to następuje specjalizacja.

Zjawisko to jest bardzo przydatne. Wyobraźmy sobie, że piszemy grę, w której występuje ruch uliczny. Na skrzyżowaniu, na czerwonym świetle stoi grupa pojazdów (samochody, rowery, motocykle). Kiedy zapali się zielone światło, wywołamy na każdym z pojazdów metodę „jedź”. Nie interesuje nas wtedy jakiego typu jest poszczególny pojazd. Wszystkie one potrafią jeździć, zatrzymywać się i skręcać.

W przypadku, kiedy ta sama grupa pojazdów dojedzie do wjazdu na autostradę, musimy zatrzymać wszystkie pojazdy, które nie są spalinowe. Jazda rowerem po autostradzie jest zakazana.

Slajd 5

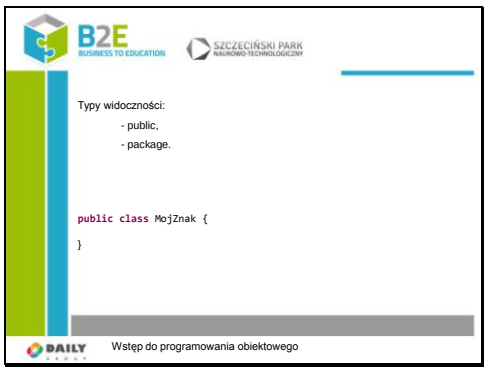


Pojedynczy obiekt może posiadać cechy, które są zbiorem cech, należących do dwóch różnych pojazdów.

Np. amfibia posiada cechy zarówno łodzi, jak i samochodu.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 6



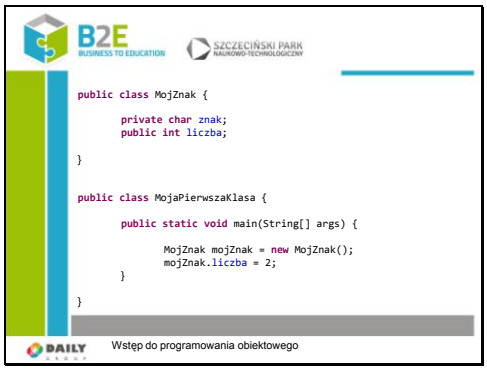
Klasę w Javie zapisujemy podając typ widoczności, słowo kluczowe „class”, nazwę klasy i nawiasy klamrowe, w których znajduje się ciało klasy.

Omówimy dwa typy widoczności: „public” i „package”.

„public” oznacza, ze nasza klasa będzie widoczna z każdego miejsca w kodzie.

Klasy w nasze aplikacji możemy trzymać w pakietach. Mechanizm działa tak, jak drzewo folderów w systemie operacyjnym. W Javie nie istnieje słowo kluczowe „package”. Jeżeli chcemy skorzystać z tego typu widoczności, to nie wpisujemy żadnego typu widoczności w definicji klasy. Wówczas nasza klasa będzie widoczna dla wszystkich klas znajdujących się w danym pakiecie.

Slajd 7



Jeżeli chcemy przechowywać w klasie pewne informacje, tworzymy pola. W Javie nie można przechowywać danych, które nie należą do żadnego obiektu!.

Pola zawierają dodatkowo typ widoczności „private”. Oznacza to, że dane zmienna jest widoczna tylko w ciele danej klasy. Pola publiczne są widoczne w każdej klasie w aplikacji.

Na slajdzie w metodzie „main” po raz pierwszy świadomie tworzymy obiekt! Obiekt jest reprezentacją danej klasy (instancją) „powołaną do życia”. Przeanalizujmy jak odbywa się ten proces.

W pierwszej kolejności musimy stworzyć referencję. Referencja jest adresem na komórkę w pamięci, gdzie znajduje się dana klasa. Referencja posiada również typ obiektu na który wskazuje.

Piszemy:

TypReferencjinazwaReferencji;

W naszym przypadku jest to referencja o nazwie „mojZnak”, typu „MojZnak”.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

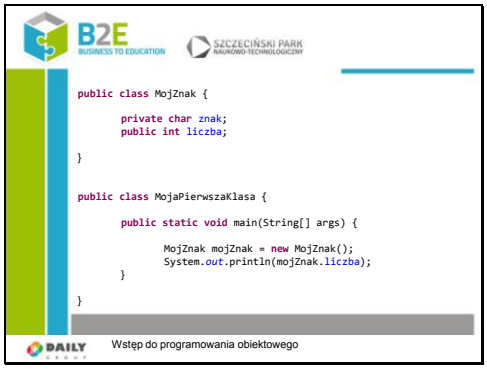
Korzystając ze słowa kluczowego „new” alokujemy obszar pamięci operacyjnej komputera na przechowanie obiektu danego typu. Następnie znajduje się konstruktor danej klasy.

Należy pamiętać, że nazwy klas piszemy dużą literą, a nazwy pól z małej.0

Do pól danej klasy odwołujemy się pisząc referencję, wstawiając symbol kropki, następnie wstawiając nazwę pola (tak jak w przykładzie).

Zauważmy, że do pola „znak” nie możemy się odwołać (jest prywatne).

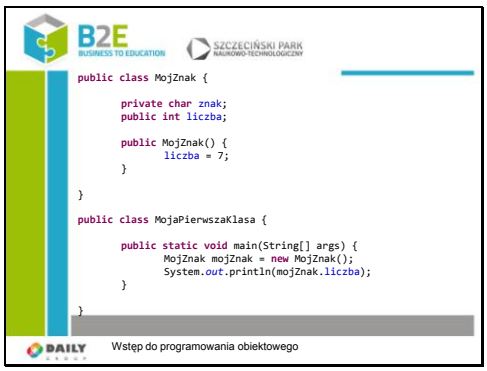
Slajd 8



W tym przykładzie chcemy wyświetlić pole „liczba” nie inicjalizując go. W tym przypadku otrzymamy 0, ale na ogół takie przypadki mogą być bardzo niebezpieczne i zagrażać stabilności naszego programu

Jeżeli tworzymy zaawansowane obiekty, musimy mieć kontrolę nad wartościami domyślnymi ich pól.

Slajd 9



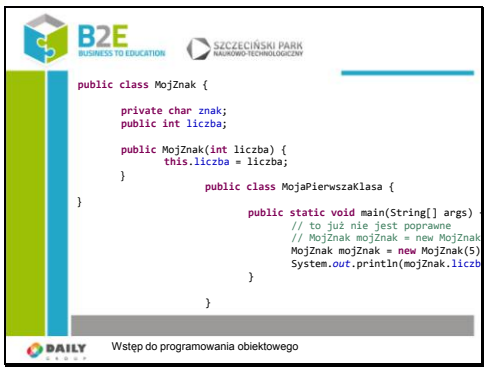
Konstruktor jest metodą specjalną. Jeżeli nie zdefiniujemy własnego, to kompilator dołącza domyślny konstruktor bezargumentowy. Dlatego w poprzednich przykładach mogliśmy pisać „newMojZnak()”.

Tworzenie własnych konstruktorów rozwiązuje problem inicjalizowania pól.

W tym przypadku, po stworzeniu klasy pole „liczba” ma wartość 7.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 10

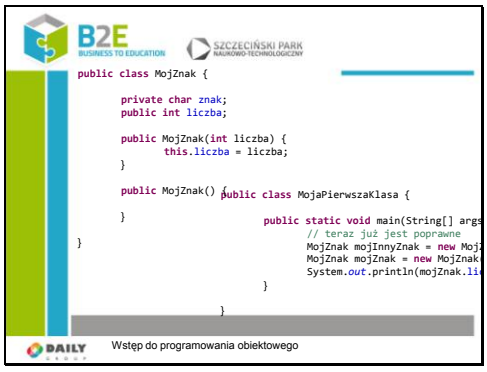


Na slajdzie widoczny jest przykład konstruktora parametrycznego. Tworząc obiekt, użytkownik musi podać wartość początkową dla pola „liczba". Kompilator nie dołączył już konstruktora domyślnego. Jest to jedyny sposób w jaki możemy stworzyć obiekt.

Słowo kluczowe „this" jest referencją na obiekt, w którym aktualnie się znajdujemy. Polecenie „this.liczba = liczba" może wyglądać dziwnie, ale jest całkowicie bezpieczne. Powoduje przypisanie wartości argumentu do pola.

W środowisku Eclipsekonstruktory możemy zbudować automatycznie korzystając z kreatora dostępnego w:
„Source/Generate Construktor using Fields..."

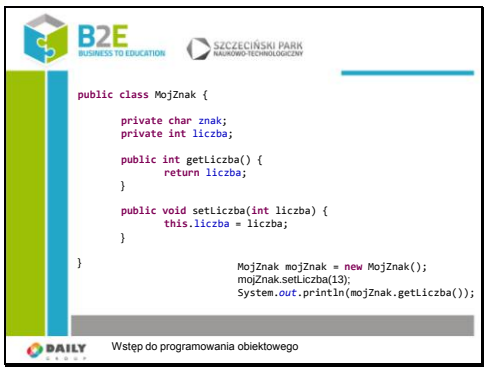
Slajd 11



Możemy stworzyć dowolną liczbę konstruktorów Muszą jednak różnić się listą argumentów, które przyjmują.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 12



Pola nie powinny mieć widoczności publicznej. Jest to zła praktyka programistyczna. Jak więc na nich operować z poziomu innych klas?

Do tego używa się metod. Metoda posiada typ widoczności, typ który zwracają, nazwę pisaną małą literą (jedyne metody, których nazwy piszemy dużą literą to konstruktory), oraz listę argumentów.

Szczególnym typem zwracanym przez metodę jest „void". Oznacza to, że metoda wykonuje jedynie pewne operacje, nie zwracając rezultatu. Tak działa metoda „setLiczba".

W innych przypadkach, jeżeli zadeklarujemy zwracanie typu „int", musimy w metodzie użyć słowa „return" wpisując za nim tę wartość. Tak działa metoda „getLiczba".

Uwaga, słowo „return" kończy wykonywanie się metody. Instrukcje, które podamy po tym słowie, nie wykonają się.

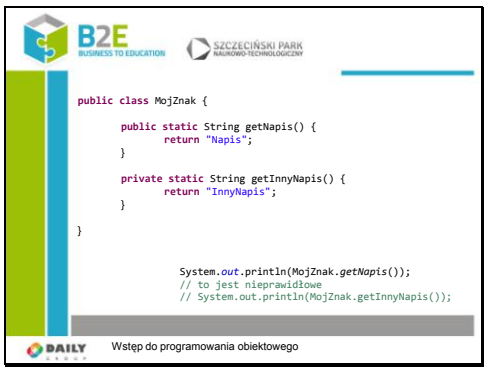
Metody zaczynające się od fraz „set" i „get" są charakterystyczne. Służą właśnie do obsługi pól.

Nie ma potrzeby ich ręcznego pisania. Można skorzystać z kreatora w Eclipse: „Source/Generate Getters and Setters..."

Metody „set" i „get" pełnią bardzo ważną role. Dają nam kontrolę nad wartościami, które użytkownik będzie chciał przypisać do pól. Np. jeżeli dane pole będzie przechowywało wartość temperatury, nie może mieć wartości mniejszej niż 273. Przy próbie zapisania wartości -400 metoda „set" powinna wpisać -273.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 13



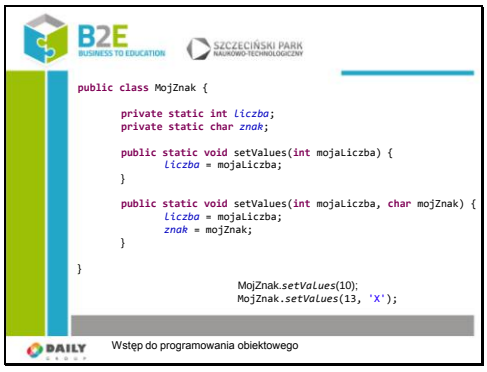
Metody, tak samo jak pola, mogą być statyczne. Oznacza to, że są takie same dla wszystkich instancji danej klasy. Nie ma też potrzeby tworzenia klasy, jeżeli chcemy skorzystać z jej elementu statycznego.

W przykładzie mamy podane prawidłowe wywołanie metody statycznej. Zauważmy, że nie mamy żadnej referencji. Metody statyczne wywoływane są bezpośrednio dla nazwy klasy.

Dlatego właśnie aplikacje w Javie uruchamia się za pomocą metody „main”. Metoda ta jest dostępna zanim zostanie stworzona instancja danego obiektu.

Oczywiście nigdy nie wywołamy metody prywatnej z poziomu innego obiektu, tak jak pokazane to jest w komentowanym kodzie.

Slajd 14



W tym przypadku nie ma sensu tworzenie instancji danej klasy, wszystko jest statyczne. Nie utworzymy również dwóch różnych obiektów typu „MojZnak”. Możemy zapisać:

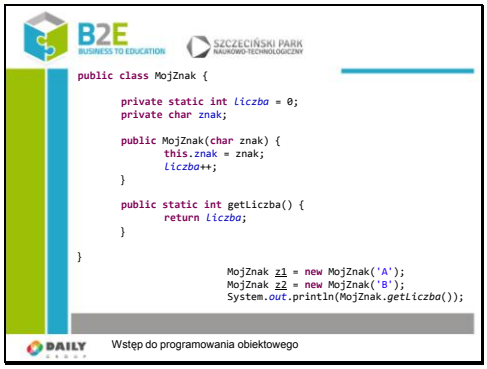
MojZnak z1 = newMojZnak();
MojZnak z2 = newMojZnak();

W praktyce jest to bez sensu. I tak nie przechowamy w obiektach różnych wartości.

Oczywiście tak, jak w przypadku konstruktorów, może być kilka metod o tej samej zazwie, różniących się listą argumnetów.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 15

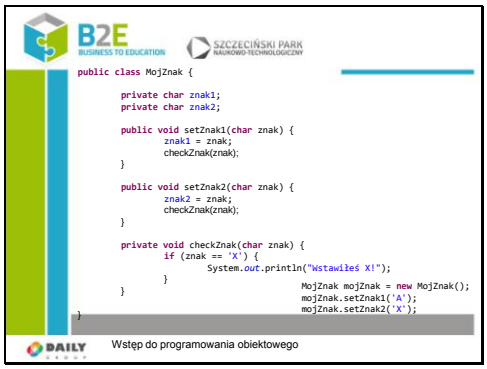


Jakie jest więc może być praktyczne zastosowanie pól statycznych? Możemy stworzyć klasę „StaleMatematyczne" i stworzyć publiczne, statyczne pole o nazwie pi i wartości „3,14". We wszystkich klasach programu będziemy mogli korzystać z tej wartości wpisując „StaleMatematyczne.pi".

Nieco ciekawszy jest przykład podany na slajdzie. Pole „liczba" jest statyczne, co oznacza, że jest wspólne dla wszystkich instancji danej klasy.

W każdym wywołaniu konstruktora następuje inkrementacja jego wartości. W prosty sposób możemy sprawdzić ile obiektów typu „MojZnak" zostało stworzonych.

Slajd 16

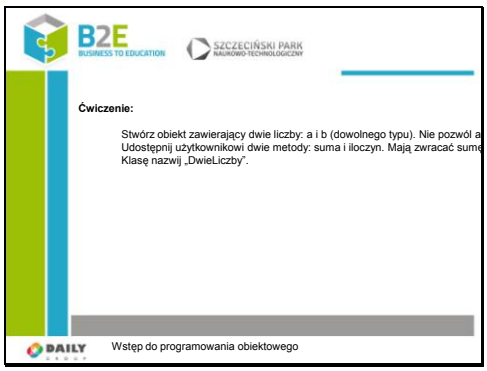


Z prywatnych metod korzystamy tylko w obrębie danej klasy. Można stosować je aby nie pisać dwa razy tego samego kodu.

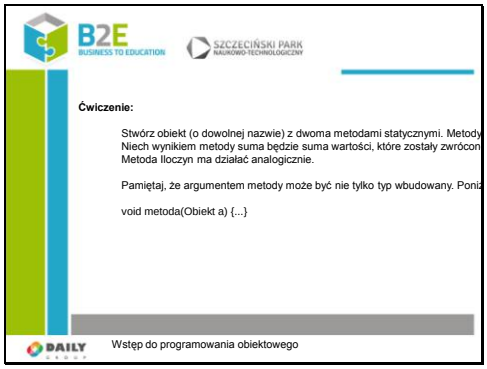
W przykładzie, w każdej metodzie „set" następuje sprawdzenie, czy ustawiana wartość jest równa „X".

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 17



Slajd 18



8.3.3 Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie będą umieli tworzyć proste obiekty w Javie, odpowiednio inicjalizować pola klas w trakcie ich tworzenia.

8.4 Lekcja 4 – Interfejsy, klasy abstrakcyjne i dziedziczenie

8.4.1 Cel lekcji

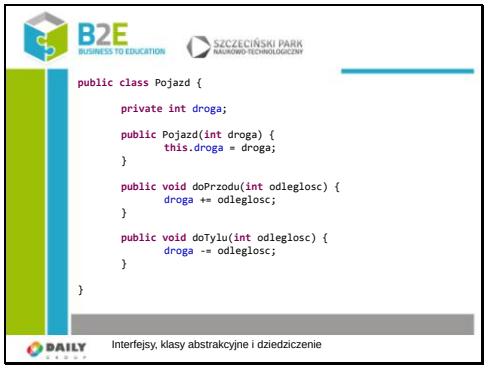
Celem lekcji jest wyjaśnienie pojęcia interfejsu i klasy abstrakcyjnej. Wytłumaczone będzie jak zwiększyć efektywność i bezpieczeństwo pracy poprzez stosowanie dziedziczenia.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

8.4.2 Treść - slajdy z opisem

Slajd 1

Slajd 2



Klasa pokazana na przykładzie implementuj w prosty sposób niektóre właściwości pojazdu.

Nowo stworzony pojazd ma licznik odległości ustawiony na zadaną wartość. Możemy poruszać się do przodu i do tyłu.

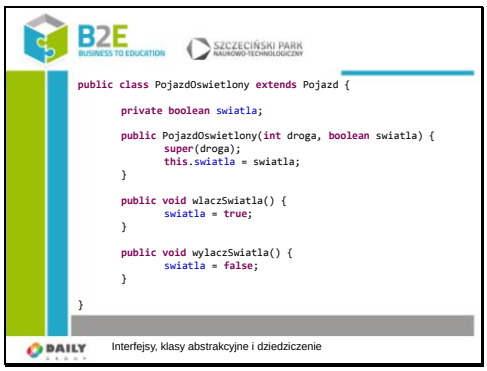
Może to być rower, motocykl , lub samochód.

Zastanówmy się co możemy zrobić, jeżeli chcemy stworzyć klasę „PojazdOswietlony”, który posiada te same właściwości co „Pojazd”. Ponadto chcemy, aby „PojazdOswietlony”, miał metody „wlaczSwiatla” i „wylaczSwiatla”.

Pisanie całości nowej klasy, przepisując część funkcjonalności z pojazdu jest niepotrzebne. W Javie (w innych językach również) nie powinno się pisać dwa razy tego fragmentu kodu. Możemy korzystać z mechanizmu dziedziczenia.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 3

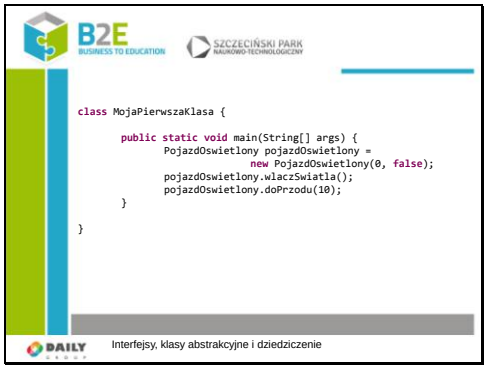


Możemy stworzyć nową klasę która posiada wszystkie właściwości klasy „Pojazd".

Po nazwie klasy, używamy słowa „extends" i podajemy nazwę klasy, po której będziemy dziedziczyć.

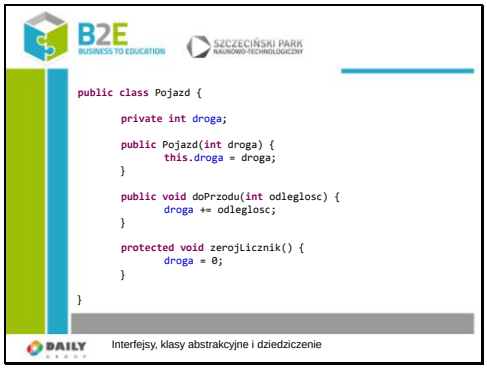
Klasa „Pojazd" posiadała tylko jeden konstruktor, z parametrem. Aby stworzyć obiekt „PojazdOswiatlony" musimy najpierw wywołać konstruktor z klasy, którą dziedziczymy. Odpowiada za to instrukcja „super". Jest to w pewnym sensie konstruktor klasy, z której dziedziczymy.

Slajd 4



Jak widać na slajdzie, na instancji obiektu „PojazdOswietlony" możemy wykonywać również metody z klasy „Pojazd". W ten sposób uniknęliśmy ponownej implementacji niektórych funkcjonalności.

Slajd 5

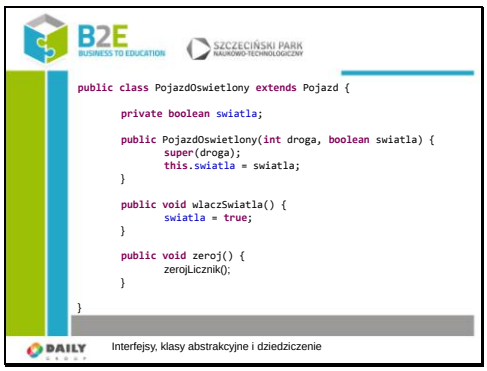


Typ widoczności „protected" oznacza to samo co domyślny typ „package". Metody i pola oznaczone w ten sposób widoczne są dla wszystkich klas znajdujących się w pakiecie.

Ponadto pola i metody mają jeszcze jedną właściwość. Są widoczne dla wszystkich klas, które dziedziczą z danej klasy.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 6

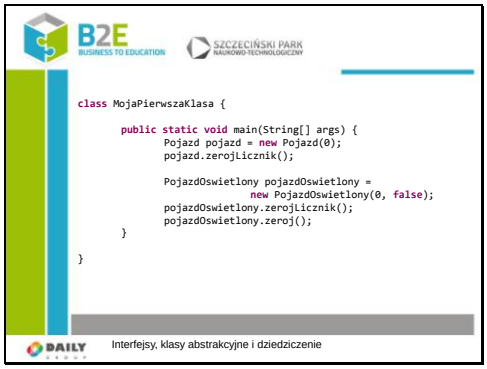


Możemy skorzystać metody „zerojLicznik” z dwóch powodów.

Po pierwsze, dziedziczymy po klasie „Pojzd”.

Po drugie, klasa „PojazdOswietlony” znajduj się w tym samym pakiecie.

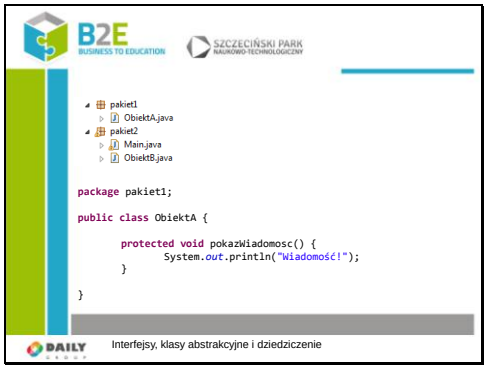
Slajd 7



Możemy skorzystać metod „zerojLicznik” tylko dlatego, że jesteśmy w tym samym pakiecie.

W innych pakietach dostępna będzie tylko metoda „zeroj”.

Slajd 8

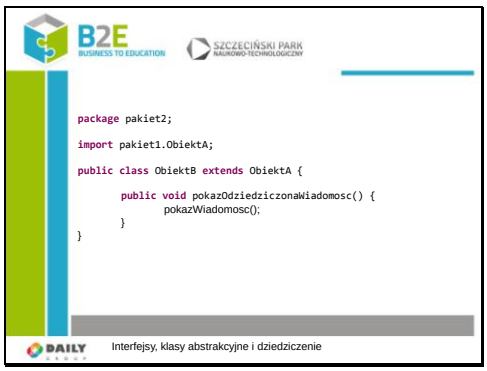


Aby lepiej pokazać działanie typu widoczności stworzone zostały dwa pakiety.

Klasa „ObiektA” znajduje się w osobnym pakiecie i zawiera metodę o właściwości „protected”.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 9

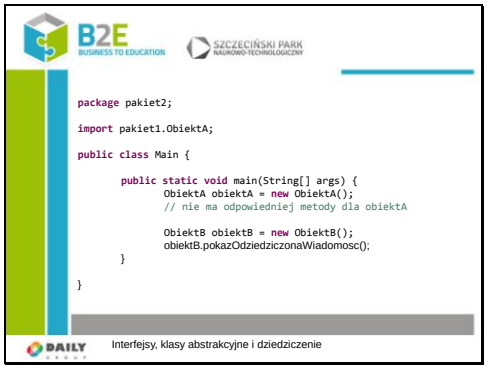


Pierwszy wers oznacza, że bieżąca klasa znajduje się w pakiecie „pakiet2”.

„ObiektA” nie jest widoczny. Musi zostać zaimportowany z innego pakietu.

„ObiektB” dziedziczy z „ObiektA”, więc ma dostęp do jego metody chronionej („protected”).

Slajd 10

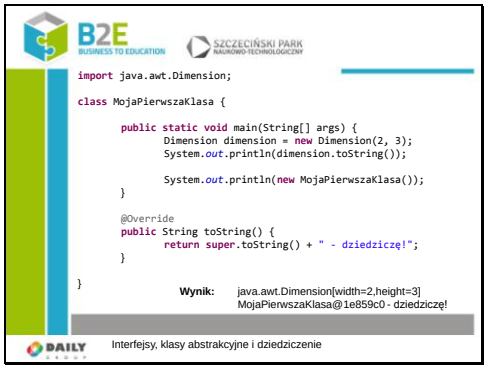


Znajdujemy się cały czas w pakiecie „pakiet2”.

Nie mamy dostępu do metody chronionej obiektu „ObiektA”.

Możemy ją jednak wywołać korzystając z „ObiektB”.

Slajd 11



Każda klasa w Javie dziedziczy po obiekcje „Object”, zawsze! „Object” posiada kilka metod i są one dostępne dla każdego stworzonego obiektu w Javie. Jedną z nich jest „toString”. Reprezentuje ona obiekt w postaci tekstu.

W metodzie „main” stworzony został obiekt typu „Dimension”. Obiekt ten przechowuje dwa podstawowe wymiary: wysokość i szerokość. Kiedy wywołamy na nim metodę „toString” otrzymamy tekstową reprezentację obiektu.

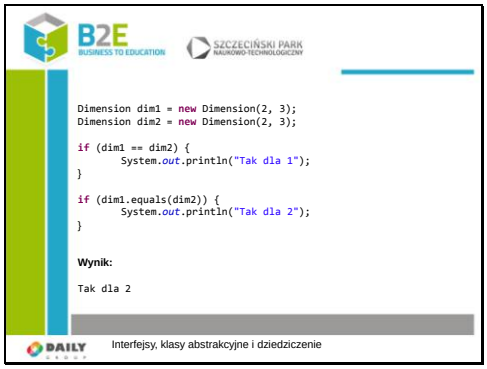
Klasa „MojPierwszaKlasa” zawiera już metodę „toString”. Chcę jednak aby zwracała wartość ustawioną przeze mnie. Wyrażenie „@Override” to adnotacja. Oznacza, że obiekt odziedziczył już taka metodę, ale chcę napisać ją we własny

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

sposób. Używając słowa „super” wywołuję metodę „toString” dla obiektu, po którym dziedziczyłem.

Metoda „println” domyślnie wywołuje metodę „toString” dla każdego obiektu, dlatego ostatnia instrukcja metody „main” działa w ten sposób.

Slajd 12



Stworzone zostały dwa z pozoru takie same obiekty. Zastanówmy się jednak. Obiekty maja takie same dane, ale zostały utworzone w niezależny sposób.

W pierwszej instrukcji „if” nie sprawdzamy równości obiektów! Sprawdzamy równość referencji. Warunek byłby prawdziwy, jeżeli obie referencje wskazywałyby na ten sam obiekt, a tak nie jest.

Dla sprawdzenia równości obiektów wywołujemy na jednym z nich metodę „equals”. Metoda ta również jest dziedziczona po typie „Object”.

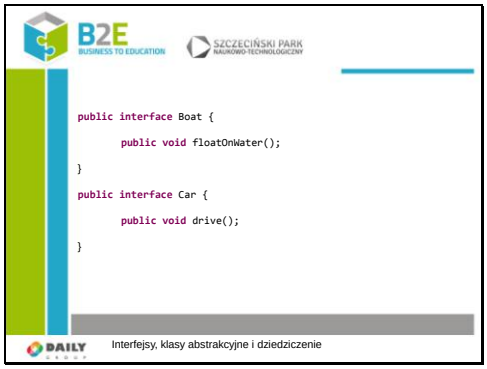
Jeżeli chcemy porównać dwie instancje stworzonej przez nas klasy, to pamiętajmy aby napisać dla niej własną wersję metody „equals”.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 13

Slajd 14

Slajd 15

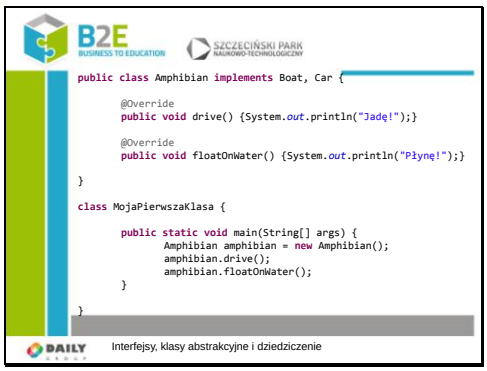


W tym celu stosuje się interfejsy. Interfejs zawiera jedynie metody bez ciał. Tak jak widzimy na slajdzie. Dodatkowo możemy wstawić listę argumentów.

Interfejs przedstawia listę operacji, które może wykonać dany obiekt.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 16



Interfejsów się nie dziedziczy. Interfejsy implementujemy,tak jak jest to pokazane w pierwszym wersie.

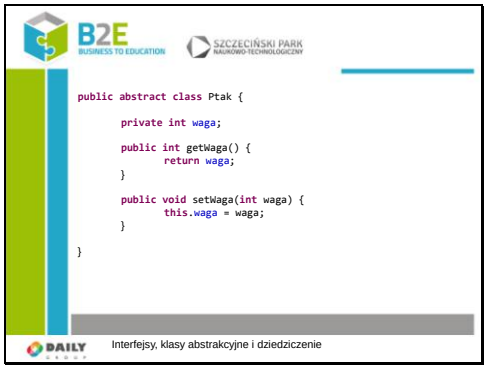
Dopiero teraz możemy nadać ciałom metody.

Slajd 17

Bieżący przykład pokazuj jaki jest sens nadawania interfejsów obiektom.

Metoda „printCar” przyjmuje jako argument obiekt typu „Car”. Obiekt „Amphibian” implementuje ten interfejs, dlatego wywołanie tej metody jest poprawne.

Slajd 18

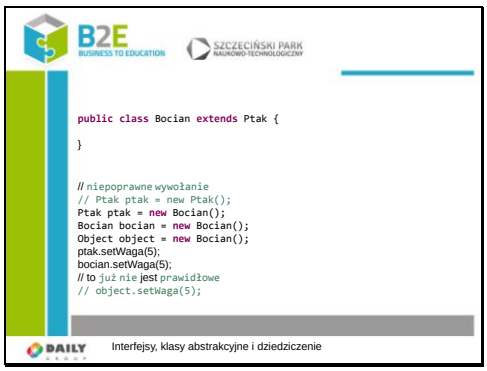


Aby stworzyć klasę abstrakcyjną należy użyć słowa „abstract” definiując klasę.

Przy jej implementacji obowiązują takie same zasady jak przy implementacji zwykłej klasy, z tą różnicą, że słowo „static” jest niedozwolone.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

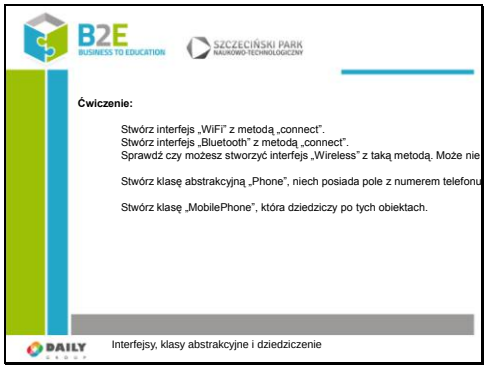
Slajd 19



Dziedziczenie po klasie abstrakcyjnej odbywa się tak samo jak po zwykłej klasie. Nie można jednak stworzyć jej instancji, tak samo jak nie można stworzyć instancji interfejsu.

Na obiekt typu „Bocian” może wskazywać referencja typu: „Object”, „Bocian” lub „Ptak”. Należy pamiętać, że na referencji typu „Object” nie wywołamy metody „setWaga”.

Slajd 20



8.4.3 Opis założonych osiągnięć ucznia

Uczeń będzie umiał wykorzystać w praktyce interfejs i klasę abstrakcyjną, tworzyć nowe klasy wykorzystując wcześniej zaimplementowany kod.

8.5 Lekcja 5 - Polimorfizm, instrukcja „final”, klasy wewnętrzne i anonimowe

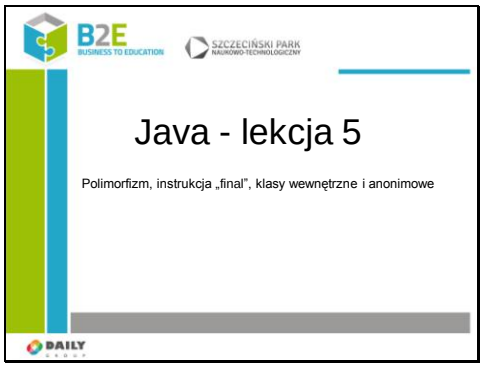
8.5.1 Cel lekcji

Celem lekcji jest wytłumaczenie pojęcia polimorfizmu, oraz jego praktycznego zastosowania w Javie. Pokazanie jak tworzyć klasy wewnętrzne, oraz efektywnie posługiwać się interfejsami.

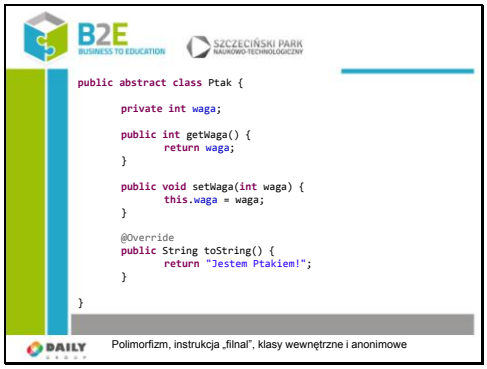
Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

8.5.2 Treść - slajdy z opisem

Slajd 1



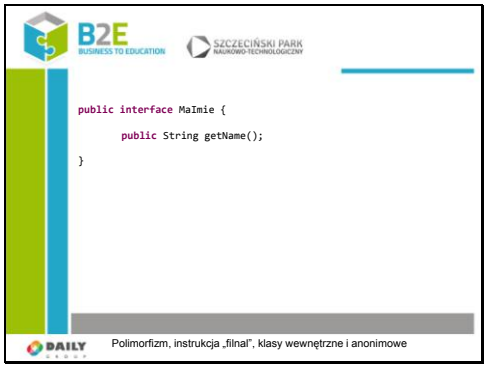
Slajd 2



```
public abstract class Ptak {

	private int waga;

	public int getWaga() {
		return waga;
	}

	public void setWaga(int waga) {
		this.waga = waga;
	}

	@Override
	public String toString() {
		return "Jestem Ptakiem!";
	}

}
```


Mamy prostą klasę "Ptak" z własną implementacją metody "toString".

Slajd 3

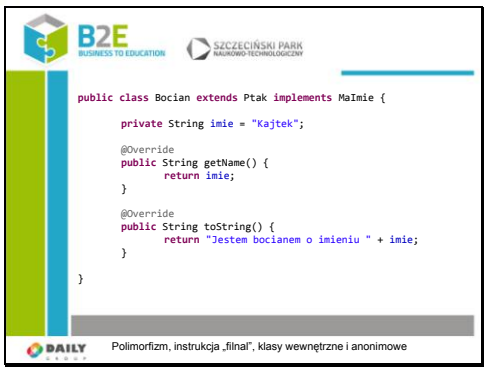


```
public interface MaImie {

	public String getName();

}
```


Istnieje również intefejs "MaImie" z metodą "getName".

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 4

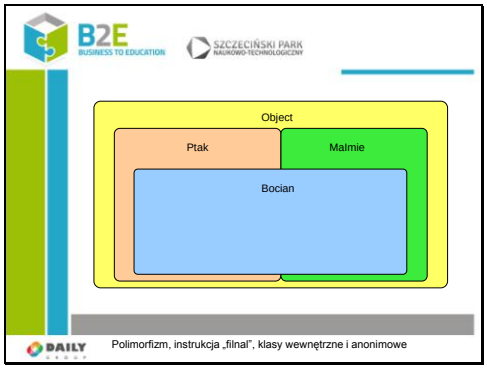


Stworzona jest klasa "Bocian", która rozszerza klasę "Ptak" i implementuje interfejs "MaImie".

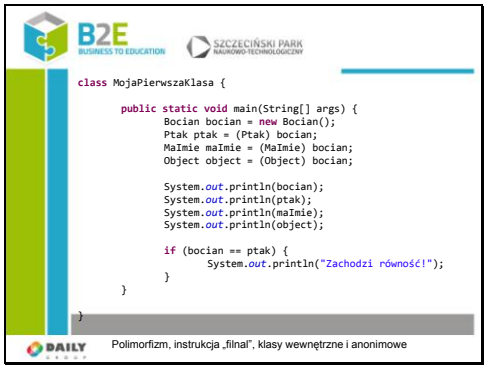
Oczywiście w przypadku implementacji interfejsu, musimy zaimplementować wszystkie metody, które do tego interfejsu należą.

Bieżąca klasa ma własną wersję metody "toString".

Slajd 5



Slajd 6



W wyniku działania programu dostaniemy cztery razy napis "Jestem bocianem o imieniu Kajtek". I napis "Zachodzi równość".

Jak widać możemy przypisać obiekt "bocian" do Referencji czterach różnych typów: "Bocian", "Ptak", "MaImie", "Object". Zjawisko to nazywamy plimorfizmem (inaczej wielopostaciowość).

Obiekt możemy traktować na różne sposoby w zależności od potrzeb.

Zastanówmy się dlaczego dla referencji typu "Ptak" nie pokazał się napis "Jestem ptakiem!"?

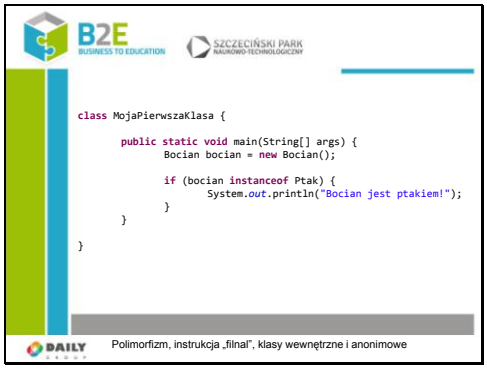
Odpowiedź może zasugerować nam porównanie. Pamiętajmy, że nie sprawdzamy równości obiektów, na które

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

wskazują referencje! Sprawdzamy, czy wskazują na ten sam obiekt. Wszystkie referencje w w metodzie "main" wskazują na obiekt stworzony przez konstruktor "Bocian()".

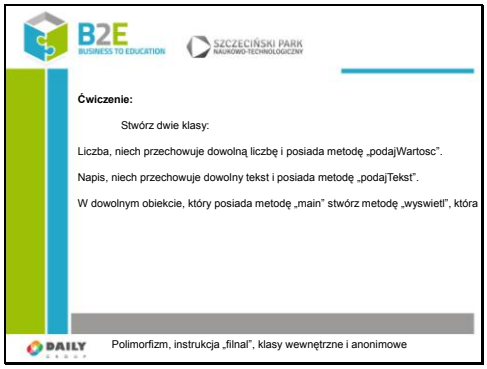
To, że metoda "toString" z klasy "Ptak" zwraca napis "Jestem ptakiem!", nie ma żadnego znaczenia. W klasie "Bocian" mamy zupełnie inną implementację tej metody. Klasa nie może posiadać dwóch metod o takiej samej nazwie i takiej samej liście argumentów.

Slajd 7



Aby móc rozpoznawać typ obiektu możemy skorzystać z operatora "instanceof". W powyższym przykładzie widzimy

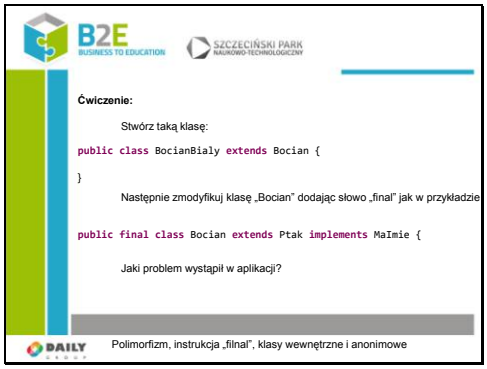
Slajd 8



Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 9

Slajd 10



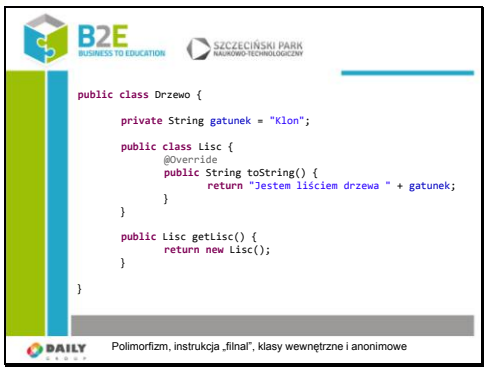
Słowo "final" użyte w definicji klasy oznacza, że nie możemy po niej dziedziczyć.

Slajd 11

Pisanie własnych wersji metod nazywamy przesłanianiem. Nie można przesłonić metody oznaczonej jako final. Jest to ostateczna implementacja.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

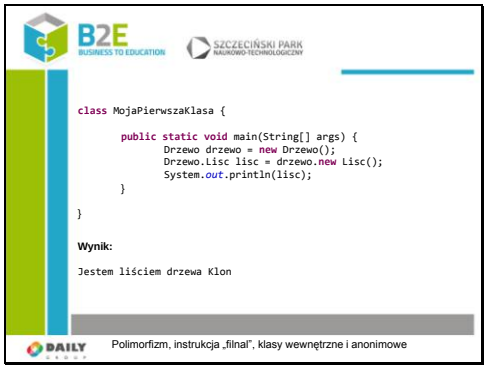
Slajd 12



Java umożliwia tworzenie klas wewnętrznych. Klasa wewnętrzna ma dostęp do elementów prywatnych klasy, w której się znajduje.

Klasa wewnętrzna może posiadać typy widoczności: prywatny, chroniony, publiczny i domyślny (pakietowy).

Slajd 13

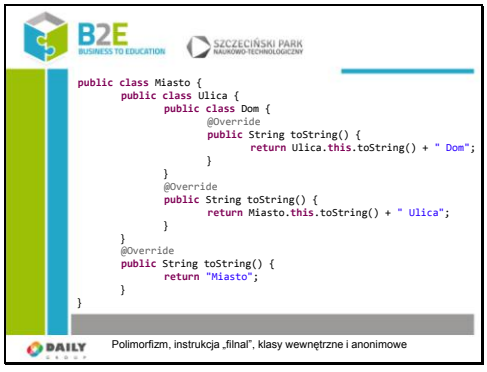


Tak wygląda prawidłowe wywołanie klasy wewnętrznej. Zwróćmy uwagę jak wygląda użycie operatora "new" dla obiektu "Lisc".

Dlaczego nie można stworzyć obiektu typu "Lisc" bez użycia obiektu "Drzewo"?

Obiekt typu "Lisc" może mieć dostęp do pól obiektu typu "Drzewo". Dlatego aby stworzyć "lisc" musimy stworzyć najpierw obiekt "drzewo".

Slajd 14



Widoczny przypadek jest nieco bardziej skomplikowany. Jak widać każda z klas może mieć własną metodę "toString".

Zwróćmy uwagę jak następuję wywołanie metody z klas zewnętrznych,

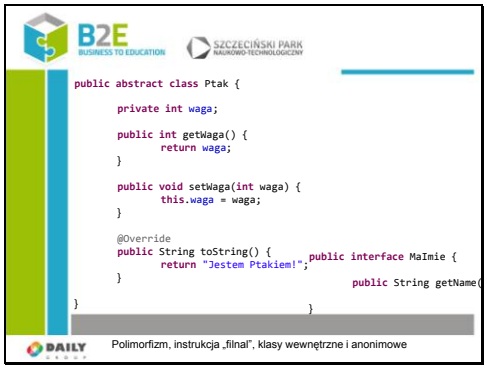
Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 15

Obiekty tworzymy dokładnie tak jak w poprzednim przykładzie.

Aby zbudować wynik wywołane zostały metody z wszystkich trzech obiektów.

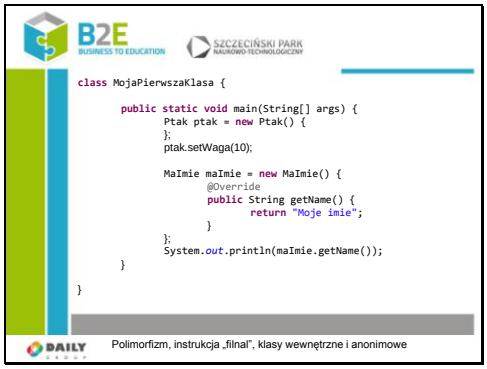
Slajd 16



Skorzystajmy z klasy abstrakcyjnej i interfejsu, których używaliśmy wcześniej.

W poprzedniej lekcji podane było, że nie możemy stworzyć instancji danej klasy abstrakcyjnej lub interfejsu. W praktyce nie możemy obejść tą zasadę tworząc klasę anonimową.

Slajd 17



"Ptak" jest klasą abstrakcyjną. W nawiasach klamrowych możemy dodać dodatkowe metody i pola do klasy. W ten sposób stworzony został obiekt anonimowy. Nie mozemy odwołać się do klasy, której obiekt jest instancją. Klasa ta nie ma nazwy, ale wiemy, że zgodnie z polimorfizmem, jest typu "Ptak".

W przypadku interfejsu sprawa jest nieco bardziej skomplikowana. Nie ma on ciał swoich metod, musimy więc je zaimplementować.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 18

8.5.3 Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie będą wiedzieć jak wykorzystuje się w praktyce polimorfizm i interfejsy.

8.6 Lekcja 6 - Typ wyliczeniowy, JavaDocs

8.6.1 Cel lekcji

Celem lekcji jest zapoznanie uczniów z typem wyliczeniowym i prostą metodą dokumentowania kodu za pomocą JavaDocs.

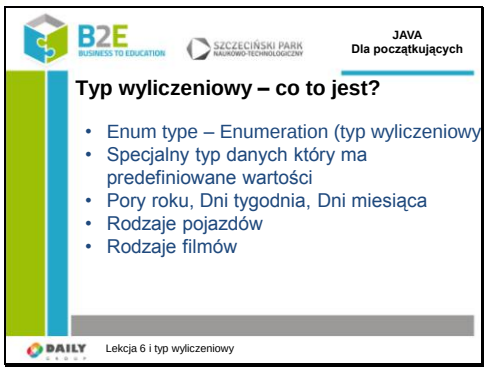
8.6.2 Treść - slajdy z opisem

Slajd 1

Na lekcji poznamy co to jest typ wyliczeniowy, do czego znajduje zastosowanie i jak można go wykorzystać w programowaniu.

Poznamy narzędzia do dokumentowania kodu

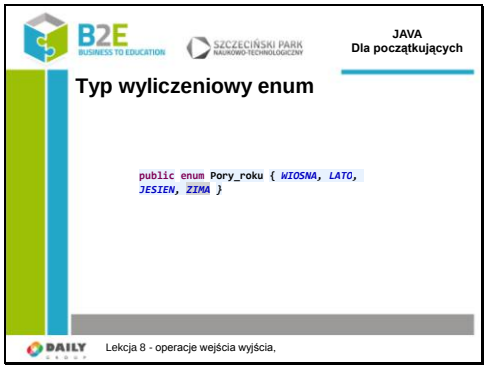
Slajd 2



Typ wyliczeniowym jest specjalny typ danych który może przyjmować jedną z zadeklarowanych stałych.

Przykładem może być pora roku. Załóżmy że chcemy aby zmienna pora_roku przyjmowała tylko ustalone przez nas wartości tj. wiosna, lato, jesien i zima.

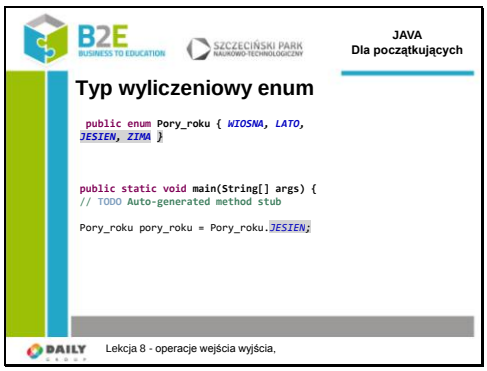
Slajd 3



Przygotujmy nową uruchamialną klasę i zadeklarujmy następujący typ zmiennej: Pory_roku który będzie przyjmować następujące wartości

public enum Pory_roku { *WIOSNA, LATO, JESIEN, ZIMA }*

Slajd 4



Przygotujmy nową uruchamialną klasę i zadeklarujmy następujący typ zmiennej: Pory_roku który będzie przyjmować następujące wartości

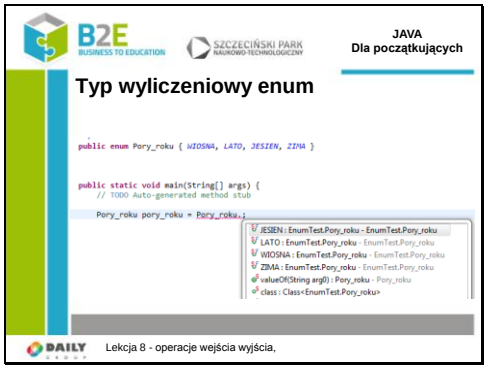
public enum Pory_roku { *WIOSNA, LATO, JESIEN, ZIMA }*

W metodzie main zadeklaruj my sobie zmienną typu Pory_roku i przypiszmy jej wartość.

Zauważmy, że wpisując wartość możemy posługiwać się zadeklarowanym typem Pory_roku.

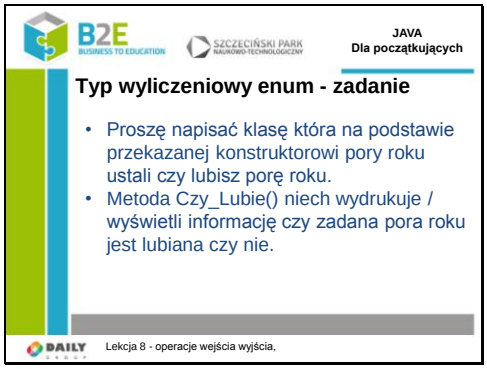
Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 5



Zauważmy, że wpisując wartość możemy posługiwać się zadeklarowanym typem Pory_roku. Eclipse podpowiada nam wartości mechanizmem intellisense.

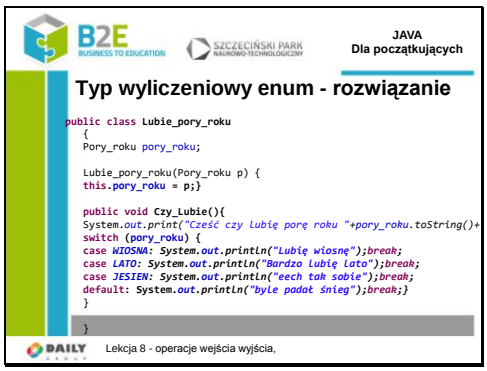
Slajd 6



Proszę wykorzystać instrukcję switch/case Dla uproszczenia klasę można zadeklarować jak zagnieżdżoną klasę programu zamiast nowego pliku dla klasy.

Inicjacja obiektu powinna następować wg przykładu:
EnumTest enumTest = **new EnumTest();**
Lubie_pory_roku w = enumTest.**new Lubie_pory_roku….**

Slajd 7

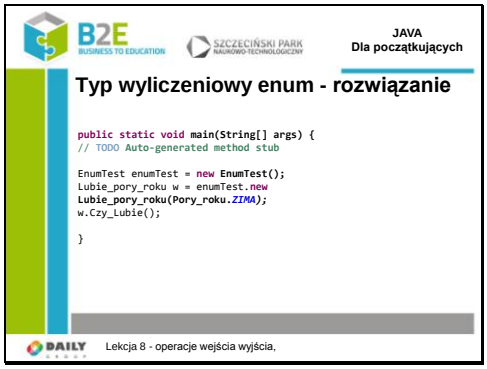


Klasa nazywa się Lubie_pory_roku
Jej konstruktor zapamiętuje zadany przy tworzeniu obiektu klasy zmienną typu Pora_roku w polu klasy.

W metodzie czy lubię zastosowano instrukcję switch / case
Proszę zauważyć jak łatwo można wykorzystać typ wyliczeniowy do obsługi poszczególnych przypadków tego typu

Dodatkowo, proszę zwrócić uwagę na konwersję wartości nazw stałych w instrukcji wyświetlającej napis „*Cześć czy lubię porę roku "+pory_roku.toString()+"?*”

Slajd 8

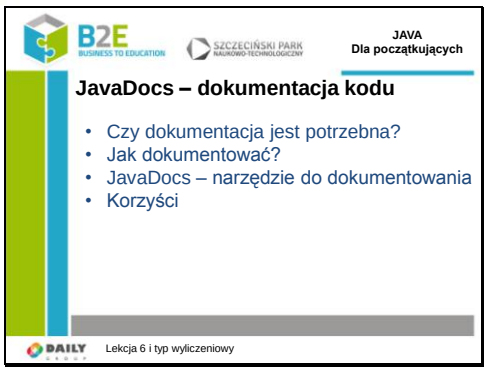


Wywołanie klasy zadeklarowanej jako klasę wewnętrzną można wykonać w zaprezentowany sposób.

Proszę zauważyć że konstruktor wołany jest ze stałą ustawianą typem Pory_roku.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 9

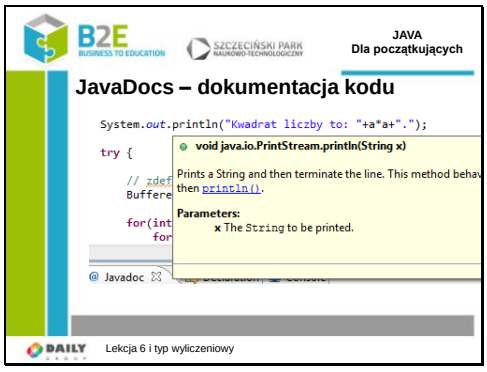


Czy dokumentacja kodu jest potrzebna?
Odp: Jest potrzebna. Wyobraźmy sobie że piszemy kod od dłuższego czasu. Powrót do napisanego kodu sprzed pół roku może okazać się trudny.
Krótki komentarz, opis zaraz przybliży nam do czego służy metoda, kto jaką wersję klasy i kiedy napisał

Jak dokumentować?
Oczywiście nie chodzi o pisanie wielostronicowych elaboratów. Opis ma być krótki, zrozumiały, zawierać tylko to co chcielibyśmy wiedzieć wykorzystując np. klasę. Najlepiej wg określonego szablonu

Narzędziem który wykorzystywany jest powszechnie jest JavaDocs. Używając środowiska Eclipse wykorzystujemy zintegrowane JavaDocs.

Slajd 10



Wystarczy że najedziemy na metodę (tutaj println) i otrzymujemy szybką informację jak dana metoda działa, jakie ma parametry

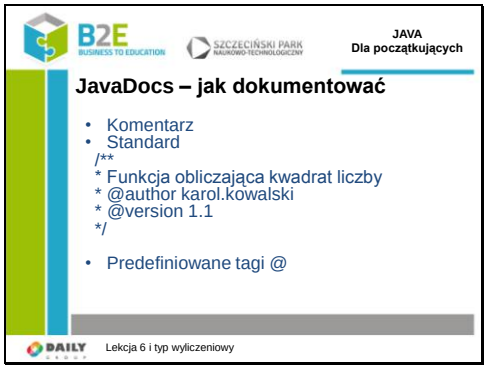
Dzięki temu nie musimy szukać informacji w dokumentacji.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 11

A oto nasz ostatni kod (klasa Lubie_pory_roku).
Po najechaniu na nazwę klasy nie wyświetlają się żadne podręczne informacje.

Slajd 12



Dokumentacja kodu jest wykonywana w formie komentarza przed typem zmiennej, klasą, metodą.

Aby odróżnić dokumentację od zwykłego komentarza stosuje się znaczniki
/**

*/

Ustalono standard opisu i tagi

Slajd 13

Proszę przyjrzeć się przykładowi który dokumentuje ostatnio zdefiniowany typ wyliczeniowy Pory_roku.

Dzięki dołączonej dokumentacji każde użycie typu Pory_roku powoduje podręczne wyświetlenie informacji.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 14

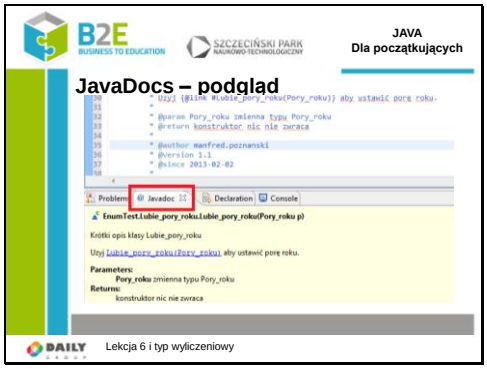
Narzędzie Eclipse wspiera przy pisaniu dokumentacji – lista dostępnych tagów

Slajd 15

Przyjęto że pierwszą linią dokumentującą będzie krótki opis co dany element wykonuje
Następnie można dodać drugą linię (po znaczniku nowej linii <p>) zawierający opis metody i jej parametrów, oraz co zwraca metoda

Potem, zależnie od uznania tagi dot. autora, wersji daty wprowadzenia.

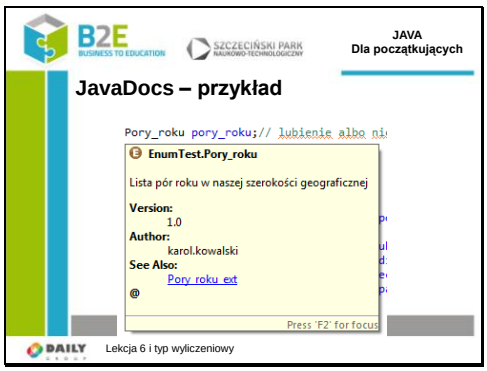
Slajd 16



Podgląd pisanej dokumentacji można śledzić na zakładce Javadoc

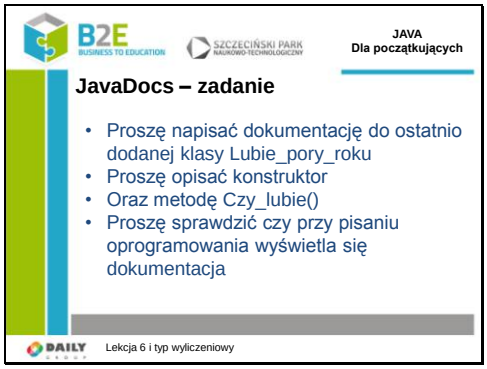
Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 17



Oto widok po najechaniu na nazwę typu Pory_roku

Slajd 18



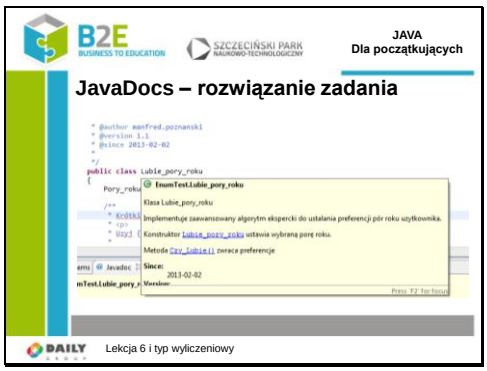
Oto widok po najechaniu na nazwę typu Pory_roku

Slajd 19

Oto przykładowa dokumentacja klasy
Proszę zwrócić uwagę na linki do poszczególnych metod klasy

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 20



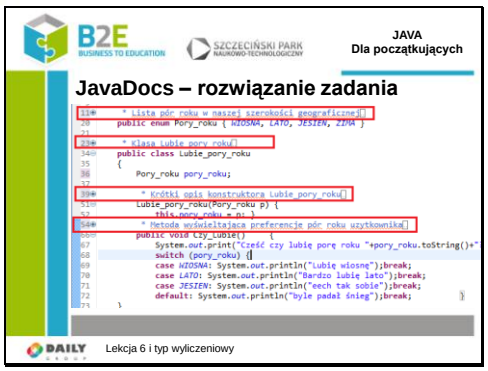
Oto przykładowa dokumentacja klasy
Proszę zwrócić uwagę na linki do poszczególnych metod klasy

Slajd 21

Slajd 22

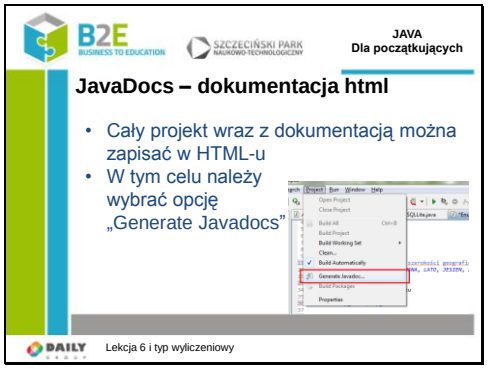
Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 23



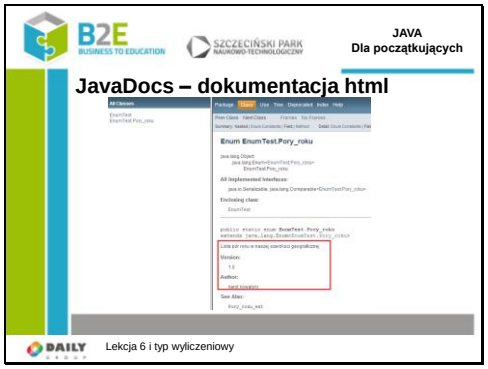
Dokumentacja nie musi zaśmiecać kodu – można ją zwinąć żeby nie przeszkadzała do jednej linii.

Slajd 24



Po uruchomieniu kreatora wybrać folder docelowy gdzie będzie zapisana strona z dokumentacją (Destination) i nacisnąć Finish

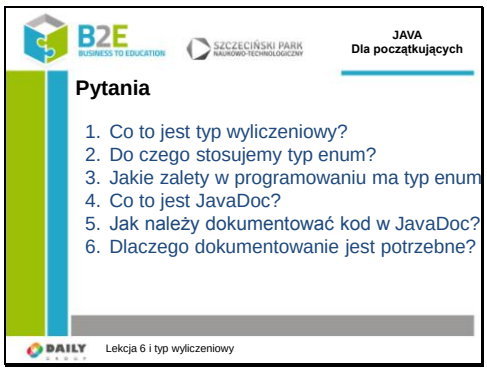
Slajd 25



Oto rezultat generowania
Automatycznie powstaje dokument HTML który przedstawia hierarchię klas, dla każdej klasy opis, listę metod, drzewo relacji i dokumentację którą wpisaliśmy

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 26



8.6.3 Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie będą umieli poznawać informacje na temat wykorzystywanych obiektów i ich metod z dokumentacji. W łatwy sposób operować typami wyliczeniowymi.

8.7 Lekcja 7 - Tablice jednowymiarowe i wielowymiarowe, kolekcje

8.7.1 Cel lekcji

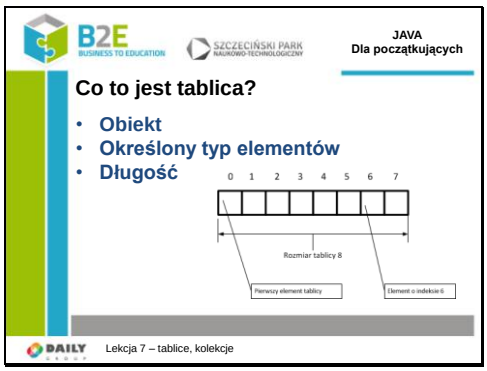
Celem lekcji jest przedstawienie podstawowych możliwości przechowywania zbiorów obiektów w tablicach, lub kolekcjach. Wytłumaczone będzie pojęcie listy.

8.7.2 Treść - slajdy z opisem

Slajd 1

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

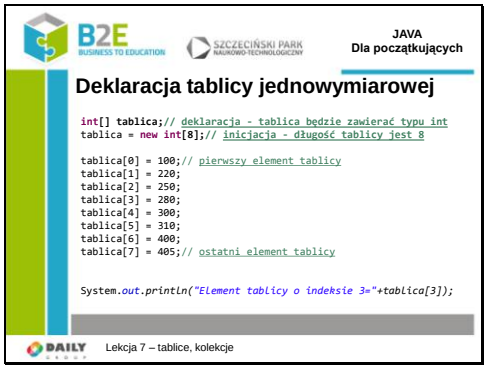
Slajd 2



Tablicą nazywamy obiekt zawierający zestaw elementu określonego typu. Długość i typ tablicy są określona przy jej deklarowaniu

Struktura taka pozwala na zarządzanie w programie zbiorem elementu tego samego typu. Np. zestaw ostatnich 10 pomiarów temperatury.

Slajd 3



int[] tablica;
W pierwszej linii wykonywana jest deklaracja tablicy. Mówi ona że:

- tablica będzie zawierać typ całkowity (int),
- nazwa obiektu tablicy to … tablica

tablica = **new int[8];**
Inicjowana jest tablica i ustawiana jest jej długość

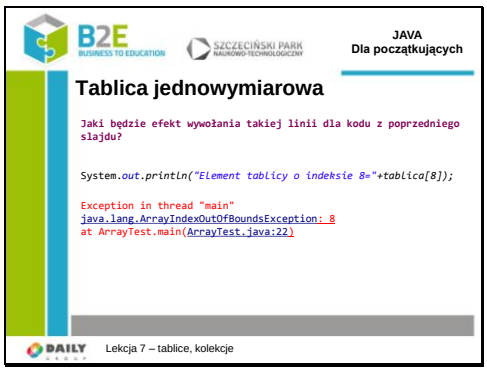
Linie zawierające „tablica[1] = 220;” ustawiają kolejne wartości tablicy

Co robi ostatnia linia kodu?
Odp. Wyświelta napis „Element tablicy o indeksie 3=280”

Co zrobić aby wyświetlić pierwszą wartość tablicy?
Odp. Ustawić indeks tablica[0]

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 4

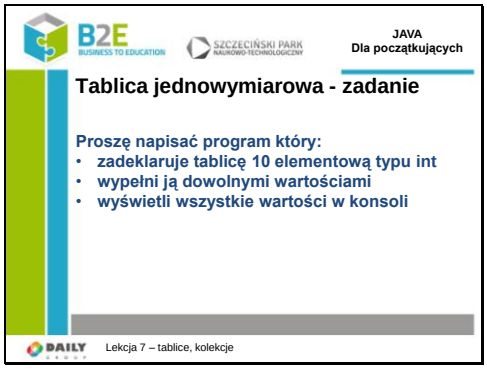


Zgłoszony zostanie błąd

Array Index Out Of Bounds Exception – Wyjątek: Indeks tablicy poza zakresem

Oznacza to, że nie wolno wychodzić poza zakres tablicy bo spowoduje to wyjątek i przerwanie działania programu.

Slajd 5



Sugestia: wykorzystać pętle for

Slajd 6

Zastosowanie pętli for i zmiennej „i” do iterowania po elementach tablicy

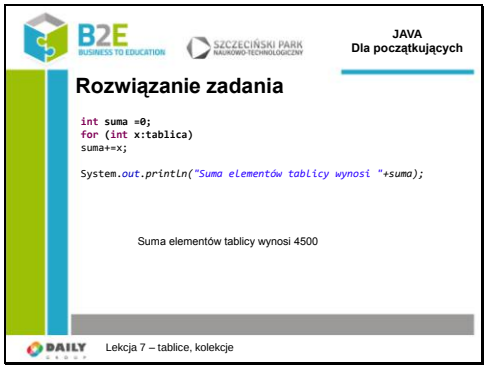
Proszę zwrócić uwagę na:
Zmienna „i” jest wykorzystana do ustalenia wartości tablicy = i*100;
Zmienna „i” jest wykorzystana do wyświetlenia indeksu przy wyświetlaniu wyników

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 7

Sugestia: wykorzystać pętle for dla wszystkich elementów tablicy
for (int x:tablica)

Slajd 8



Zastosowanie pętli for dla wszystkich elementów tablicy nie wymaga iteratora (zmiennej „i”).

Proszę zwrócić uwagę na formuę suma+=x;
Co ona oznacza?
Odp: suma = suma + x;

Slajd 9

Inny sposób deklarowania i inicjowania tablicy – przyda się tylko wówczas gdy wartości tablicy będą w programie stałe.

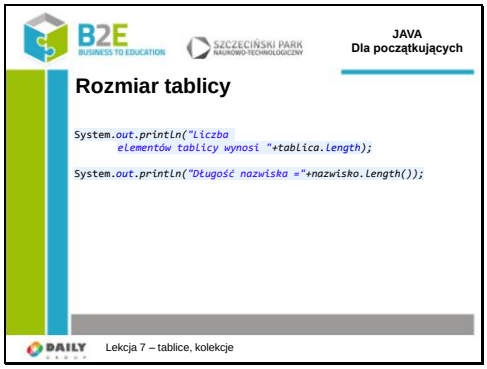
Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 10

Łańcuch znaków String jest tablicą jednowymiarową zawierającą elementy typu znak (char).

Proszę sprawdźcie jak będzie działał powyższy program.
Odp. Jak się można spodziewać program wyświetli nazwisko „Kowalski”.

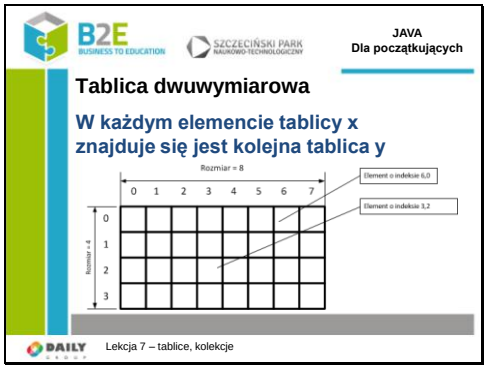
Slajd 11



Typ String jest specyficznym typem tablicowym i posiada metody do wykonywania operacji na znakach. Metoda length() zwraca długość napisu zapamiętanego w zmiennej.

Typ tablicowy posiada licznik .length który zwraca liczbę elementów w tablicy

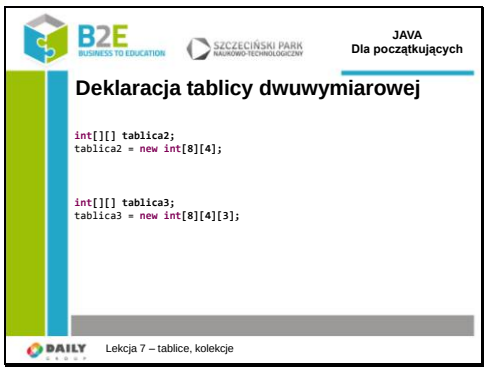
Slajd 12



Rozważmy tablicę dwuwymiarową 8x4
Jak będzie wyglądała deklaracja takiej tablicy?

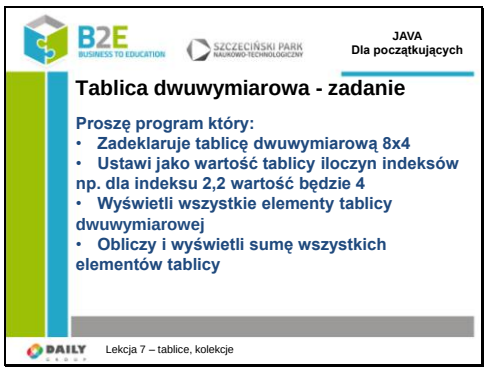
Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 13



Jak zadeklarować tablicę trójwymiarową 8x4x3?

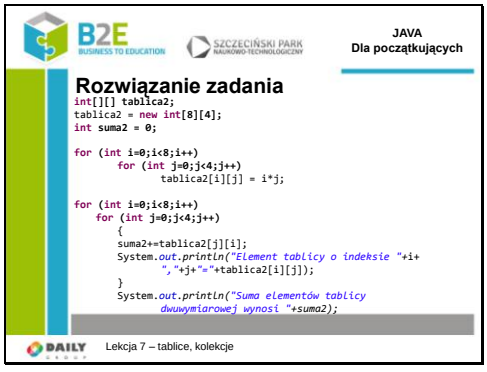
Slajd 14



Sugestia: wykorzystać 2x pętle for dla wszystkich elementów tablicy w wymiarze x i y (8,4)

Kto poda pierwszy wynik sumy?
Odp: 168

Slajd 15



W rozwiązaniu zadania jest błąd.

Kto wskaże w którym miejscu, na czym błąd polega i co stałoby się przy uruchomieniu tego programu.

Należy bardzo uważać żeby nie pomylić kolejności iteratorów

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

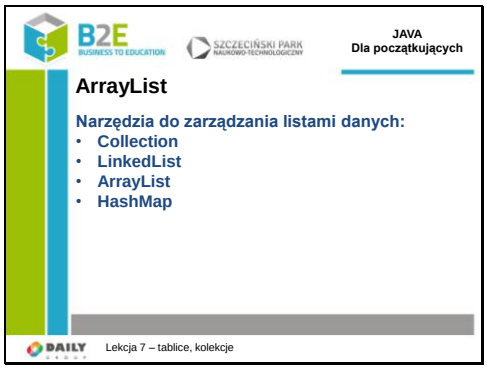
Slajd 16

Zarządzanie tablicami z przykładów jest na dłuższą metę dość uciążliwe.
Wady:

- Musimy pilnować zakresów,
- Musimy deklarować typy obiektów,
- Aby wyszukać dany element musimy napisać parę linii kodu,
- …

W celu wyeliminowania tych ale również innych wad środowisko programistów przygotowało sobie specjalne narzędzia. Są to struktury pozwalające na łatwe zarządzanie danymi.

Slajd 17



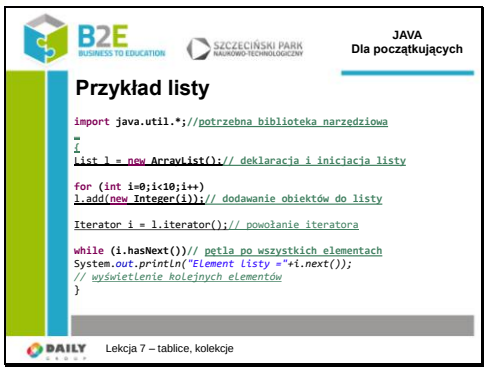
Zarządzanie tablicami z przykładów jest na dłuższą metę dość uciążliwe.
Wady:

- Musimy pilnować zakresów,
- Musimy deklarować typy obiektów,
- Aby wyszukać dany element musimy napisać parę linii kodu,
- …

W celu wyeliminowania tych ale również innych wad środowisko programistów przygotowało sobie specjalne narzędzia. Są to struktury pozwalające na łatwe zarządzanie danymi.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 18



Pierwsza uwaga do przykładu – należy wykorzystać bibliotekę narzędzi java.util. Dla uproszczenia dodamy wszystkie narzędzia w domenie util poprzez instrukcję

import java.util.*; - gwiazdka oznacza że wszystkie biblioteki w danej bibliotece będą dołączone do programu.

Alternatywnie można dodać następujące linie:
import java.util.ArrayList;
import java.util.List;
import java.util.Iterator;

Deklaracja listy o nazwie „l" jest prosta:**List l = new ArrayList();**
Warto zauważyć że nie deklarowaliśmy typu tablicy.

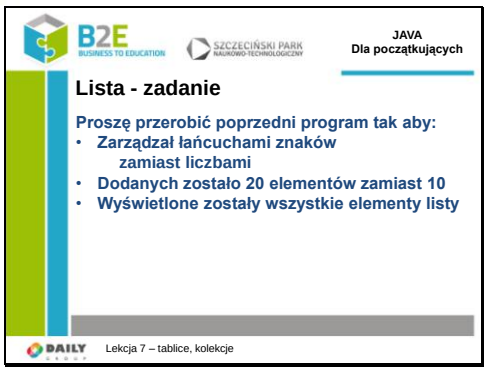
Następnie dodawane są **obiekty** do listy (10 elementów). Aby powołać obiekt stosujemy instrukcję new Integer(<wartość>) – nowy obiekt typu Integer.

Aby wyświetlić wszystkie elementy z listy należy powołać iterator. Jest to obiekt pozwalający na dostęp do kolejnych elementów listy.

Zastosowano pętlę while której sprawdza czy jest na iteratorze kolejny element. Jeśli tak instrukcja i.next() zwraca obiekt i jednocześnie przechodzi do kolejnego elementu listy.

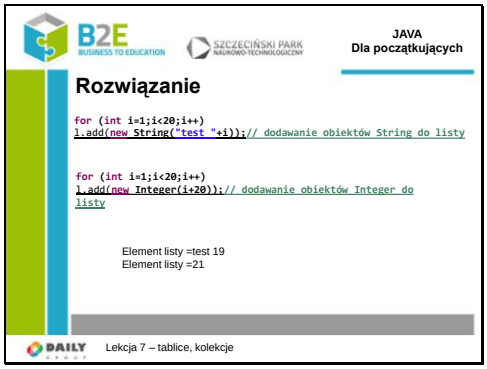
Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 19



Sugestia: wykorzystać obiekt String analogicznie jak Integer

Slajd 20



Wystarczy w jednym miejscu zmienić instrukcję powołującą nowe obiekty

Z new Integer(i) na np. new String(„tekst”+i)

Co warto zauważyć:

Przerobienie programu jest dużo prostsze niż w przypadku tablic [][]

Jedno miejsce

Nie musimy zajmować się kontrolą liczby elementów

Nic nie stoi na przeszkodzie aby lista zarządzała różnymi obiektami np. String i Integer

Mamy do dyspozycji całą paletę narzędzi do zarządzania listą – następny slajd

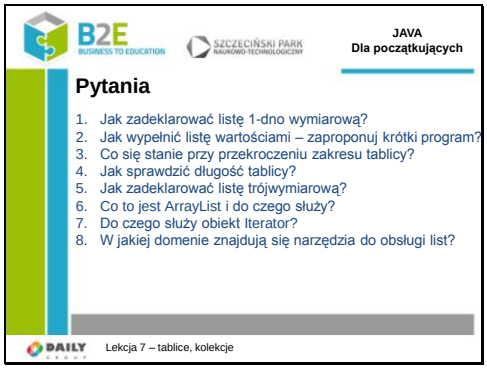
Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 21

Są wśród nich:

- Dodawanie i wstawianie obiektów (add), na pozycję, dodawanie całej kolekcji
- Czyszczenie listy
- Wyszukiwanie obiektów
- Pobieranie obiektów wg nr
- Usuwanie wg indeksów, wg obiektów, wszystkich
- Tworzenie podlist
- Ustawianie obiektu na pozycji
- Sprawdzanie rozmiaru
- Iterowanie

Slajd 22



8.7.3 Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie będą umieli gromadzić obiekty w tablicach i kolekcjach. Posiądą również umiejętność przeglądanie tych struktur za pomocą pętli.

8.8 Lekcja 8 - Operacje wejścia – wyjścia i wyjątki

8.8.1 Cel lekcji

Celem lekcji prezentacja możliwości zabezpieczenia aplikacji przed błędami, oraz umiejętność wymiany danych z zasobami z poza programu (sieć, pliki lokalne).

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

8.8.2 Treść - slajdy z opisem

Slajd 1

Slajd 2

Każdy program powinien mieć możliwość komunikowania się z użytkownikiem, systemem operacyjnym czy innymi aplikacjami.

W języku Java komunikacja ta jest realizowana za pomocą strumieni. Obsługa strumieni jest zaimplementowana w domenie **system.io** za pomocą obiektów out i in.
import java.io.*

Przykładem jest często wykorzystywana metoda print lub println wyświetlająca napis w konsoli
System.out.println("Zła liczba „);

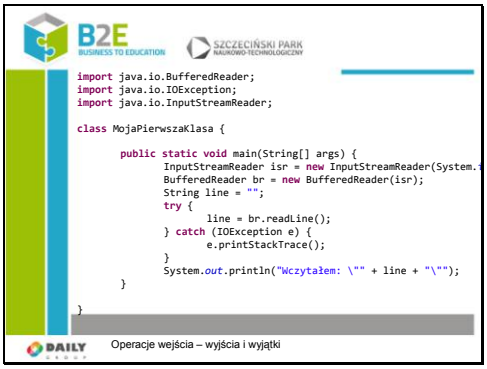
Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 3

Przykładem użycia strumienia wyjściowego out jest wyświetlenie tekstu w konsoli.
W powyższym przykładzie program wyśle do strumienia zewnętrznego tekst.

Przykład wykorzystuje standardowy strumień out zdefiniowany w obiekcie System.

Slajd 4

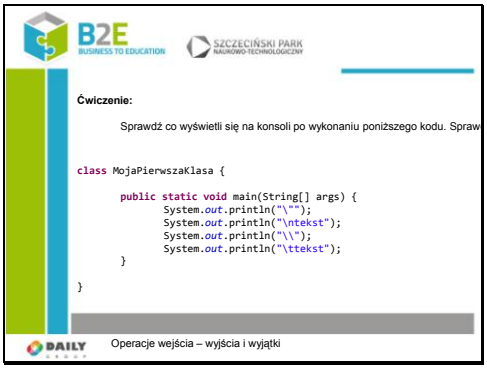


Operacje wejści-wyjścia w Javie polegają na obsłudze strumieni. Maszyna wirtualna udostępnia obiekty, które, np. odczytują znaki z klawiatury.

Przykład podany na slajdzie odczytuje cały wers wpisany do konsoli.

Pojawił się nowy element: bloki "try-catch". Jest to mechanizm obsługi błędów. W przypadku gdybyśmy utracili połączenie z urządzeniem z którego czytamy, program jest w stanie obsłużyć błąd i zareagować w odpowiedni sposób. Uchroni to naszą aplikacje od nieprzewidzianego zachowania.

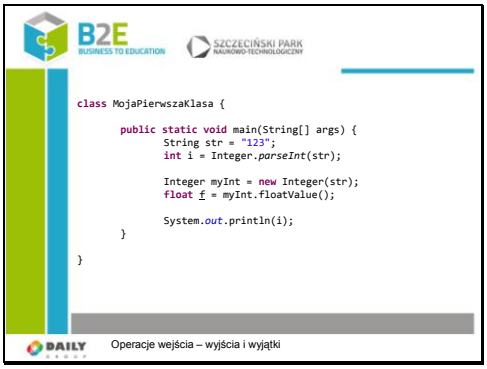
Slajd 5



\" - cudzysłów
\n – znak nowej linii
\\ - znak \
\t - tabulator

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 6



Każdemu typowi wbudowanemu odpowiada Klasa. W praktyce możemy stosować obiekty i typy wbudowane zamiennie. Nie ma to znaczenia dla obliczeń. Możemy również stosować operatory matematyczne, tak samo, jak dla typów wbudowanych.

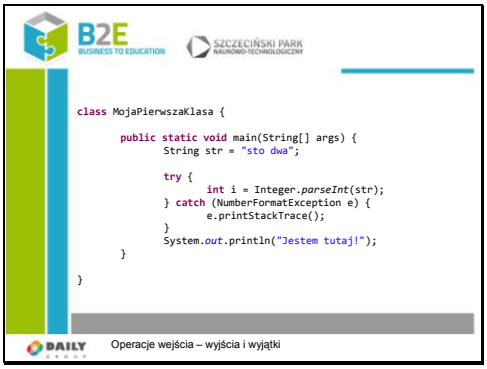
Po przypisaniu nowego obiektu typu "Integer" do referencji "myInt" możemy sprawdzić jakie metody należą do tej klasy. W tym celu po wpisaniu nazwy i kropki wciskamy "Ctrl" + "Spacja". Środowisko Eclipse poda nam automatycznie listę metod. Możemy poruszać się po niej "góra-dół" za pomocą strzałek. Zobaczymy również opisy każdej z metod. Wykonując ćwiczenia sprawdzajmy czym są obiekty, których używamy.

Slajd 7

M

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 8

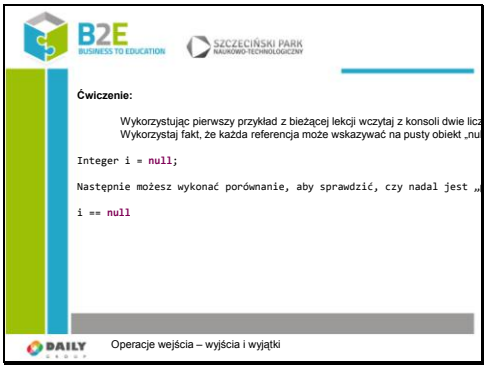


W przypadku próby stworzenia instancji klasy "Integer", korzystając z nieprawidłowego łańcucha znaków, program zakończy się niepowodzeniem.

Możemy tego uniknąć stosując blok "try-catch", tak jak na przykładzie. W bloku "try" wykonują się polecenia potencjalnie "niebezpieczne". Oczywiście łańcuch znaków "sto dwa" jest nieprawidłowy i przejdziemy do bloku "catch". Problem zostanie przechwycony i wykonają się polecenie z tego bloku. Na ekranie zobaczymy cały stos programu. W przyszłości, w dużych aplikacjach, bedziemy mogli łatwo odszukać klasę, metodę, a nawet wers, w którym pojawił się błąd, wraz z opisem błędu.

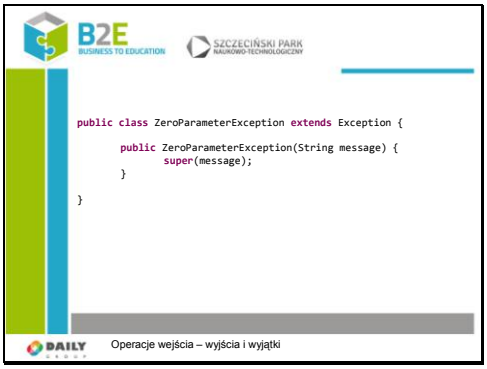
Fakt, że w konsoli pokazał się raport o błędzie, nie wpływa na niestabilność programu. Napis "Jestem tutaj!" również będzie widoczny. Uniknęliśmy niespodziewanego przerwania programu.

Slajd 9



Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

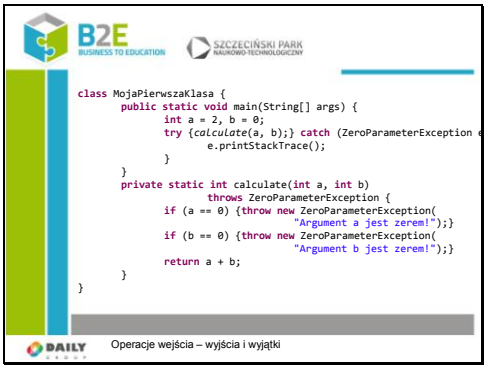
Slajd 10



Stworzone przez nas klasy mogą powodować charakterystyczne tylko dla nich błędy. Warto, aby w takim przypadku rzucały wyjątek stworzony specjanie dla nich.

Stworzenie takiego wyjątku jest bardzo proste. Wystarczy dziedziczyć po klasie "Exception" i zaimplementowć przynajmniej jeden konstruktor.

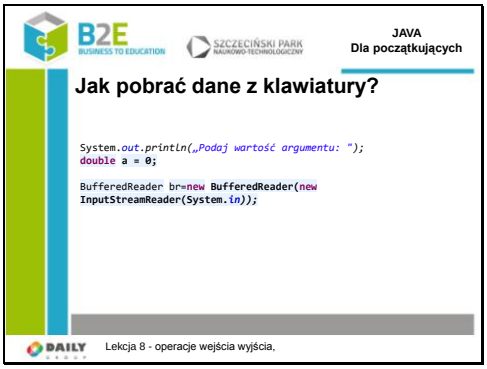
Slajd 11



Jeżeli metoda może wyjątek wyrzucić, należy zadeklarować to za listą argumentów, tak jak na slajdzie.

Kiedy zajdzie niepożądana sytuacja (w tym przypadku jeden z argumentów będzie zerem), wyjątek powinien być wyrzucony. Użycie "throw" kończy wywołanie metody. Nie zwróci ona żadnej wartości.

Slajd 12



W celu odczytu danych z klawiatury należy przygotować urządzenie wejściowe.
Przykładowa deklaracja może wyglądać następująco:

```
BufferedReader br=new BufferedReader(new InputStreamReader(System.in));
```

Obiekt br typu BufferReader będzie czytać wprost ze strumienia wejściowego System.in

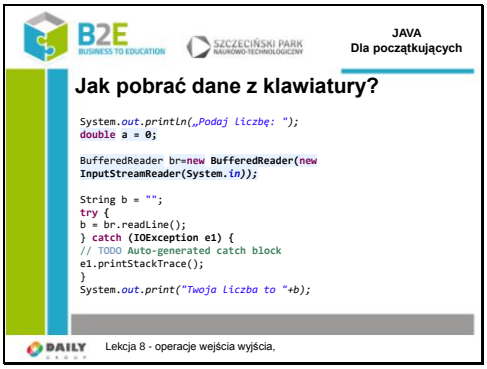
Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 13

Aby wykorzystać klasę BufferReader i InputStremReader należy użyć bibliotek java.io
W tym celu na samym początku kodu należy dodać linię:

```
import java.io.*;
```

Slajd 14



Jakiego typu są odczytane dane z klawiatury?
Odp: Są to znaki, w szczególności ciąg znaków czyli typ String.

Sprawdźmy czy działa – wprowadźmy ciąg znaków z klawiatury, po zatwierdzeniu enterem niech wyświetli się wprowadzona wartość.

Proszę dodajcie linię:

```
String b = "";
b = br.readLine();
```

Exclipse automatycznie zwróci uwagę, że br.readLine(); należy otoczyć sekcją krytyczna.

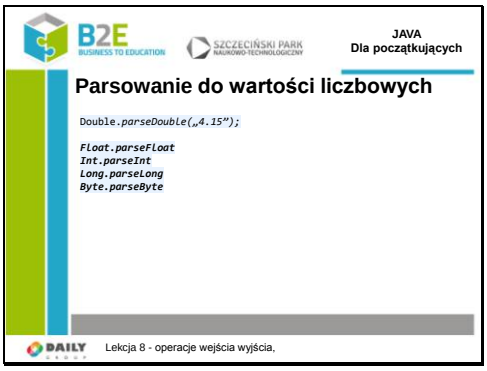
Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Wszystkim działa?

Pytanie: Co należy zrobić aby program obliczył potęgę wpisanej przez użytkownika liczby?
Odp:

- Skonwertować ją do typu liczbowego double
- Wykonać operację mnożenia

Slajd 15



Powyższa formuła dokonuje parsowania. Podobnie można zrobić dla innych typów

Uwaga!
Co zwróci formuła w przypadku gdy zadana wartość nie będzie nadawała się do konwersji na liczbę?

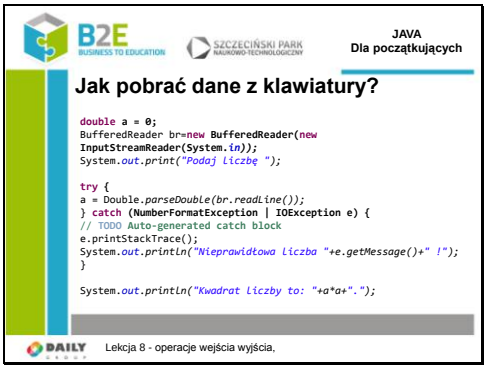
Oczywiście wygenerowany zostanie wyjątek. Dlatego należy instrukcję również otoczyć sekcją krytyczną try/catch

Pytanie:
Jak należy dokończyć nasz program?

- Parsowanie
- Obsługa błędu parsowania – sekcja catch
- Obliczenie a*a
- Wyświetlenie

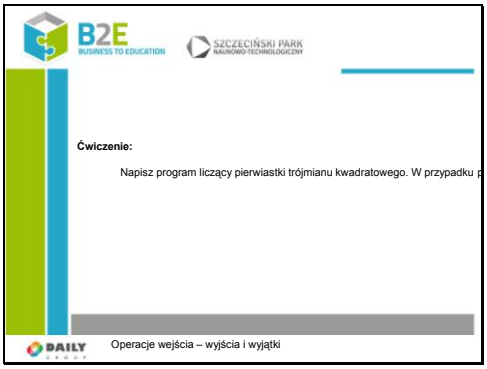
Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 16



Program powinien wyglądać mniej więcej tak:

Slajd 17



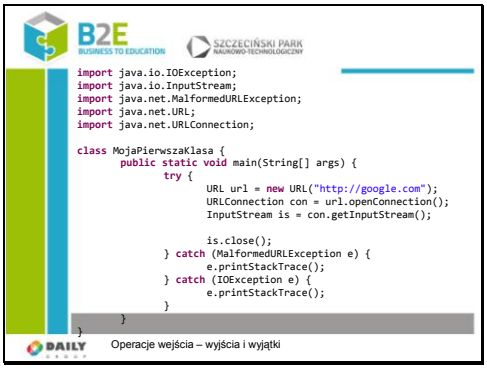
Człowiek - najlepsza inwestycja

KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY

Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 18



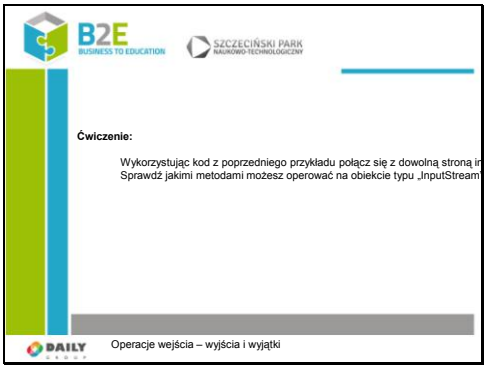
Strumienie są bardzo uniwersalne. Java umożliwia w łatwy sposób obsługę sieci za ich pomocą.

Spójrz na wyjątki i spróbuj odgadnąć jakie problemy mogą wystąpić w trakcie połączenia z internetem.

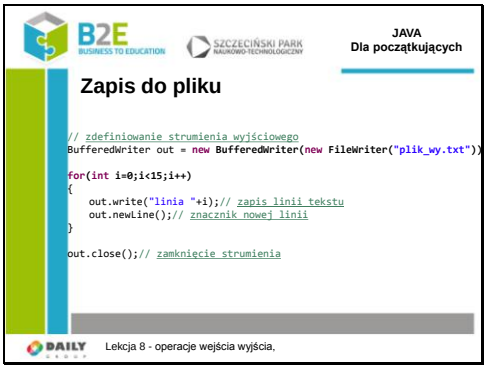
Nieprawidłowy adres URL i błąd odczytu (zerwane połączenie).

Odpowiednie nazwnictwo czyni pracę programisty łatwiejszą.

Slajd 19



Slajd 20



Przykładem użycia strumienia wyjściowego out jest zapis tekstu do pliku.
Jak widać jest on znacznie bardziej skomplikowany niż przykład wyświetlania tekstu na ekranie.

Pierwszym etapem jest zdefiniowanie obiektu out powiązanego z plikiem „plik_wy.txt”.

Następnie w 15 iteracjach zapisywane są kolejne linie do pliku – np. tekst („linia 1”)

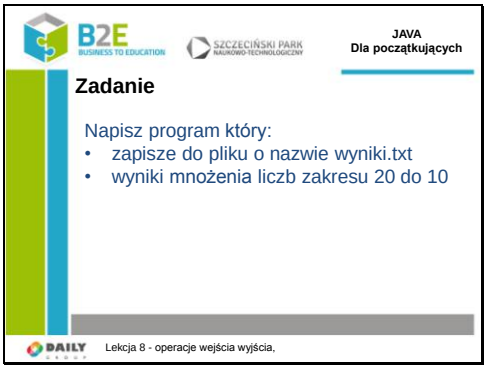
Ostatecznie następuje zamknięcie strumienia out.close();

Uwaga:
Aby skorzystać z klas BufferedWriter,

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

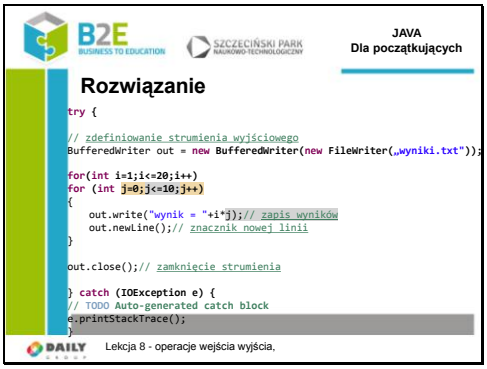
FileWriter należy zaimportować bibliotekę **import java.io.***

Slajd 21



Zadanie nie powinno trwać dłużej niż 10 minut
Należy przypomnieć o lekcji związanej z sekcjami krytycznymi try/catch

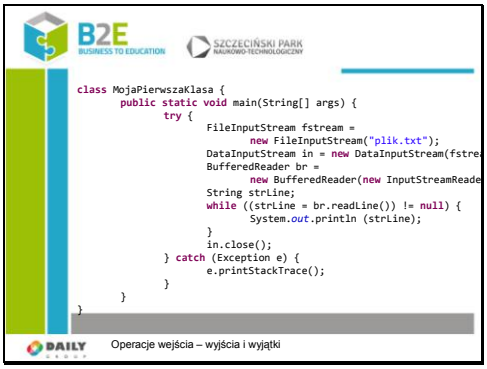
Slajd 22



Uwaga,
System zgłosi potrzebę zastosowania sekcji krytycznej (try/catch) do obsługi ew. błędów IO

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

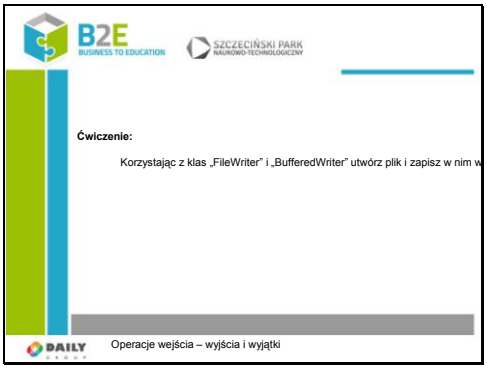
Slajd 23



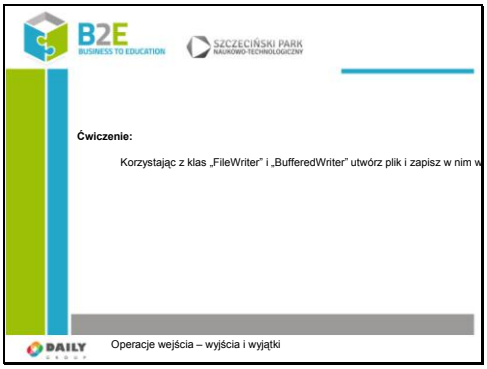
Podstawową operacją na strumieniach jest obsługa plików. Przykład pokazuje jak odczytać plik.

Domyślna lokalizacja pliku jest w folderze naszego projektu.

Slajd 24

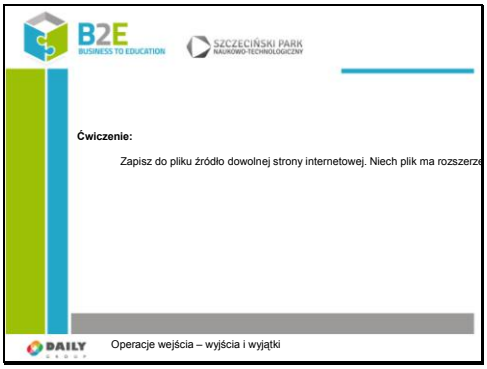


Slajd 25



Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 26



8.8.3 Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie będą skorzystać z połączenia sieciowego i plików lokalnych w komputerze. Ponadto będą umieli zabezpieczyć swój program przed potencjalnymi błędami.

8.9 Lekcja 9 - Graficzny interfejs użytkownika - Swing

8.9.1 Cel lekcji

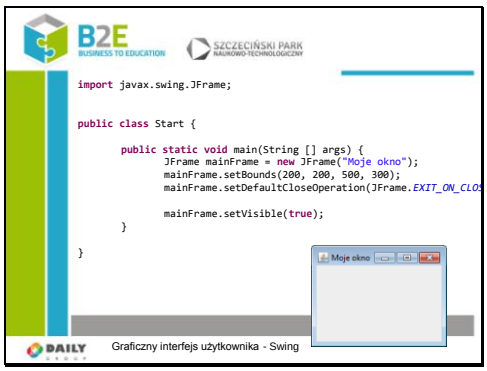
Celem lekcji jest pokazanie możliwości graficznego pakietu Swing. Wytłumaczenie jak poprawnie korzystać z tych elementów, oraz jak zarządzać ich rozkładem w oknie.

8.9.2 Treść - slajdy z opisem

Slajd 1

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 2



Okno możemy stworzyć korzystając z klasy "Jframe".

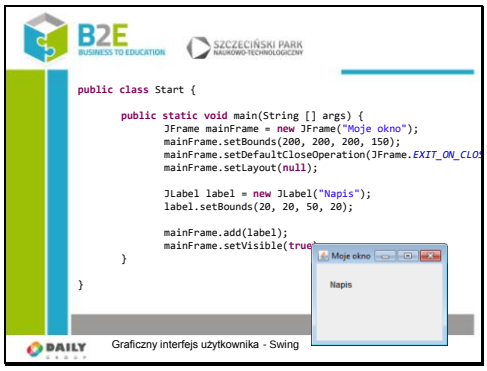
Łańcuch znaków podany w konstruktorze jest tytułem okna.

Metoda "setBounds" posiada 4 argumenty:
- odległość w poziomie od lewego, górneo rogu,
- odległość w pionie od lewego, górneo rogu,
- szerokość okna,
- wysokość okna.

Jeżeli nie ustawimy metody "setDefaultCloseOperation", to kliknięcie przycisku zamknięcia w prawym, górnym rogu nie spowoduje wyłączenia programu. Okno zniknie, ale aplikacja nadal będzie chodziła w tle!

Ostatnie polecenie powoduje wyświetlenie się okna.

Slajd 3



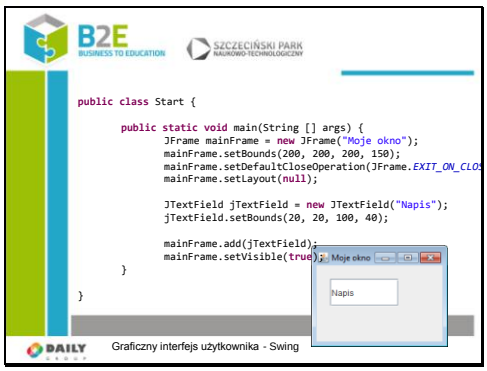
Do dyspozycji mamy wiele gotowych komponentów. Jeżeli chcemy wyświetlić tekst możemy skorzystać z klasy "Jlabel".

Skorzystajmy z metody "setLayout" z argumentem "null". Będziemy wtedy mogli skorzystać z absolutnego pozycjonowania komponentów widoku. W późniejszych slajdach zostanie wytłumaczone, jak układać elementy korzystając z zarządców rozkładu (layoutów).

Nie zapomnijmy dodać obiektu "label" do okna.

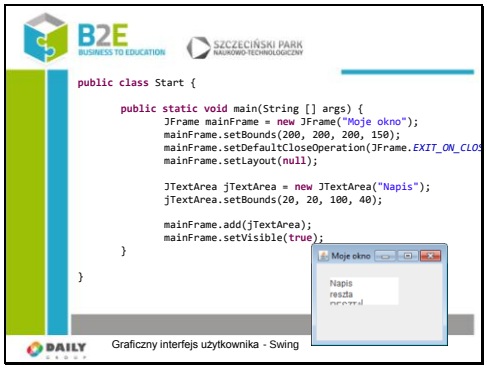
Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 4



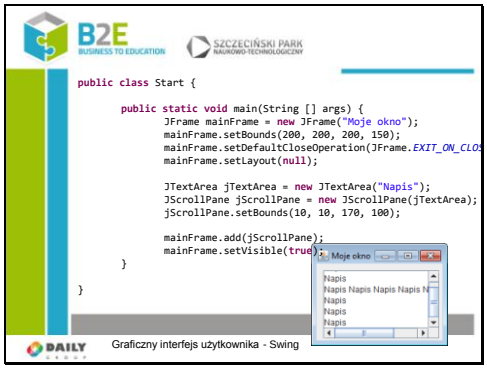
Klasa "JtextField" umożliwia wyświetlenie pola tekstowego z możliwością edycji. Ma ono jednak ograniczenie. Tekst może mieć tylko jeden wers.

Slajd 5



"JTextArea" umożliwia wyświetlenie tekstu składającego się z wielu wersów. Zauważmy jednak, że tekst nie mieści się w polu.

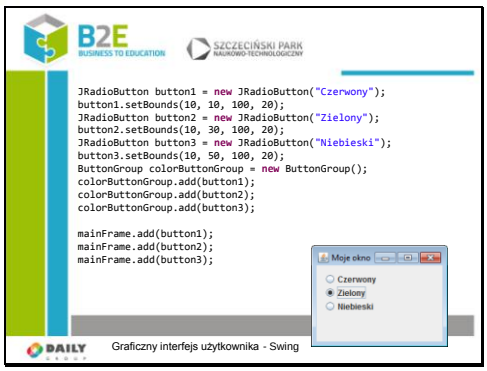
Slajd 6



Problem z widocznością tekstu możemy rozwiącać za pomocą "JScrollPane". Obiekt "JTextArea" może teraz zmieniać swój rozmiar dynamicznie. Przeglądanie ułatwiają paski przewijania.

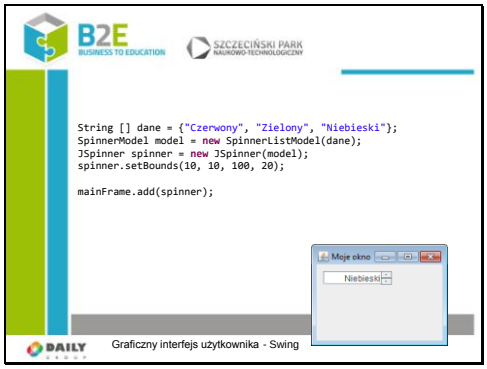
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 7



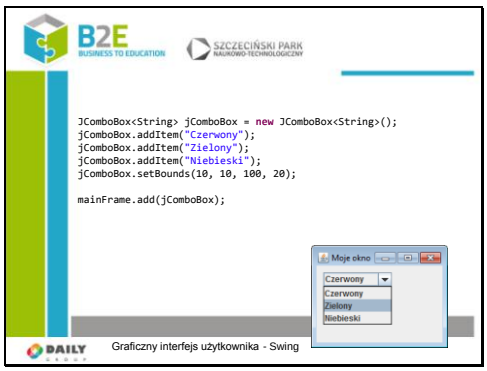
Obiekty "JRadioButton" umożliwiają wybór pojedynczej opcji. Zaznaczony może być tylko jeden element dlatego wszystkie muszą tworzyć grupę. Obiekt "ButtonGroup" kontroluje wszystkie komponenty "JRadioButton". Jeżeli nie użylibyśmy go, opisywana zasada nie zachodziłaby.

Slajd 8



Obiekt typu "JSpinner" również umożliwia nam wybór tylko jednej opcji, ale w nieco inny sposób. Zwróćmy uwagę na sposób, w jaki wstawiamy dane do obiektu. Korzystamy z modelu. Wynika to z zastosowania Wzorca projektowego MVC przez twórców Swingu. Wzorzec ten rozdziela warstwę widoku od warstwy danych.

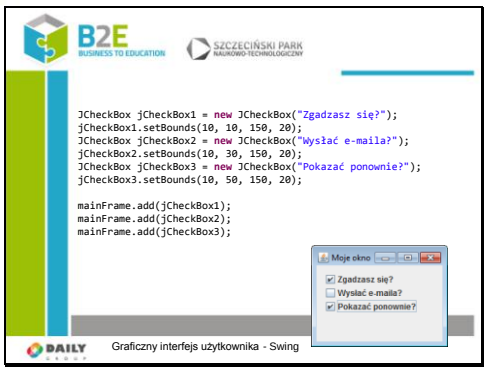
Slajd 9



"JComboBox", też pozwala wybrać tylko jedną opcję. Jest on typem generycznym. W nawiasach "<>" podajemy typ obiektu jaki będzie przechowywany w "JComboBox".

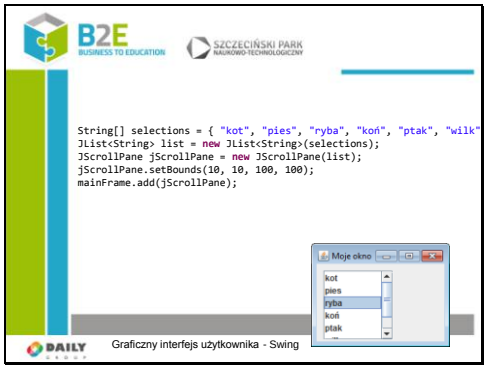
Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 10



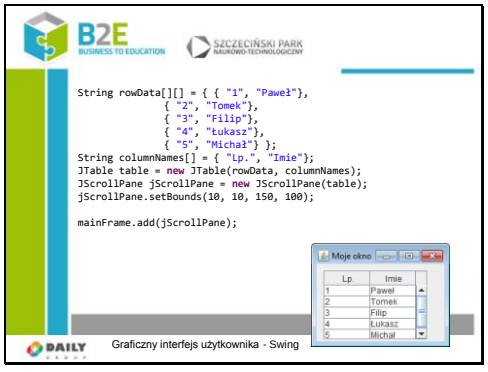
"JCheckBox" umożliwia wybór kilku opcji. Elementy te sa całkowicie niezależne. Nie dodajemy ich do żadnej grupy, tak jak to było w przypadku "JRadioButton".

Slajd 11



"JList" jest również typem generycznym. Pozwala on wyświetlić dane w bardziej czytelny sposób. Można ustawić możliwość zaznaczania pojedynczego lub wielokrotnego.

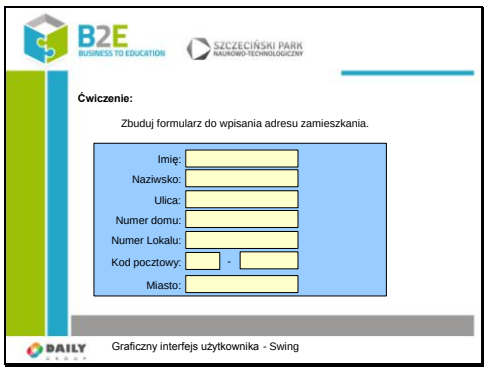
Slajd 12



"Jtable" działa podobnie jak "Jlist", z tą różnicą, że umożliwia wyświetlenie dodatkowych kolumn.

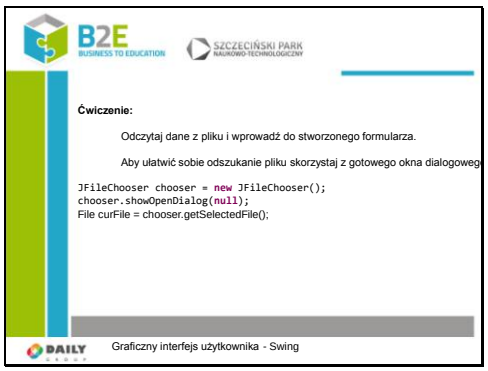
Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 13



Slajd 14

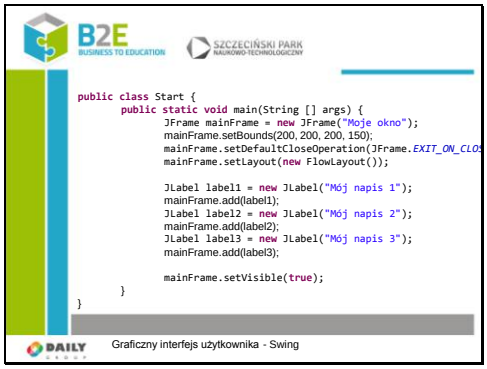
Slajd 15



Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

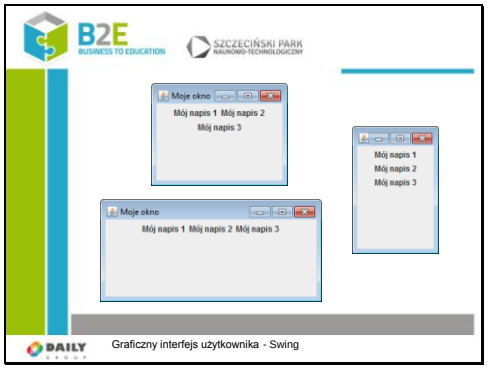
Slajd 16

Slajd 17



We wcześniejszych przykładach do metody "setLayout" wstawialiśmy obiekt "null". Teraz skorzystamy z zarządcy rozkładu o nazwie "FlowLayout". Nie musimy już układać każdego elementu osobno. Co więcej, nawet gdy użyjemy metody "setBounds", nie uzyskamy oczekiwanego efektu. "FlowLayout" decyduje o położeniu komponentów.

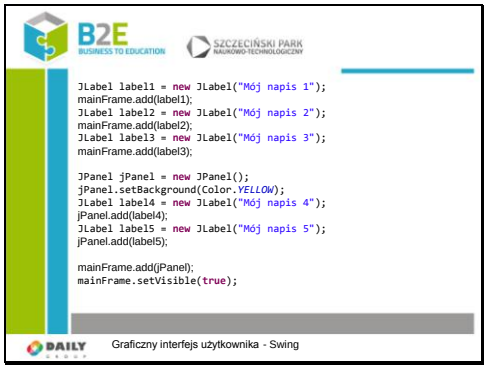
Slajd 18



Możemy uruchomić poprzedni przykład i zobaczyć jak "FlowLayout" zarządza widokiem.

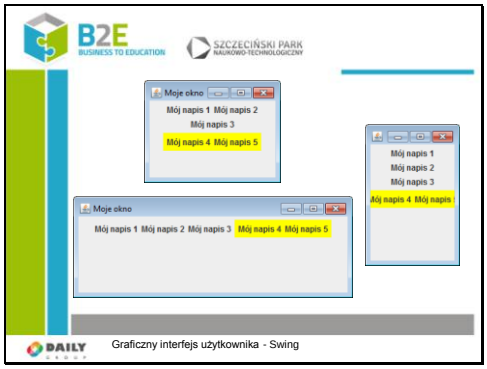
Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 19



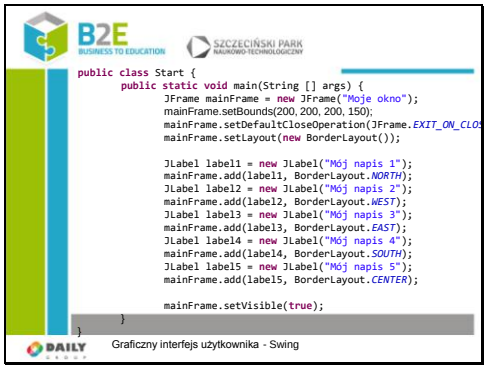
Korzystając z klasy "JPanel" możemy grupować elementy. "Mój napis 4" i "Mój napis 5" będą znajdować się wewnątrz panelu.

Slajd 20



Możemy zaobserwować, że przemieszczany jest cały panel. Dzieje się tak, ponieważ "JPanel" ma własnego zarządcę rozkładu i ustawiony "FlowLayout" nie zarządza komponentami z żółtego pola.

Slajd 21



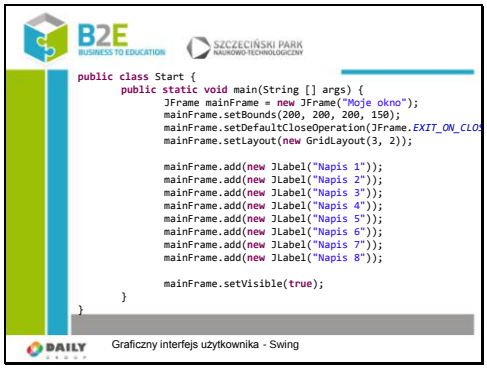
Kolejnym przykładem jest "BorderLayout". Jeżeli nie wywołamy metody "setLayout", jest on domyslnym zarządcą. Dodając każdy element podajemy dodatkowo jeden z parametrów: "NORTH", "SOUTH", "WEST", "EAST" i "CENTER".

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 22

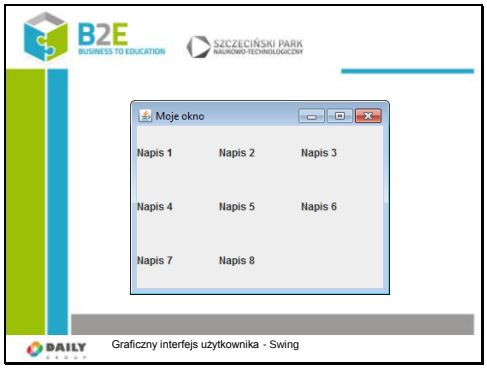
Zmieniając rozmiar okna możemy zaobserwować gdzie poszczególne obszary się znajdują.

Slajd 23

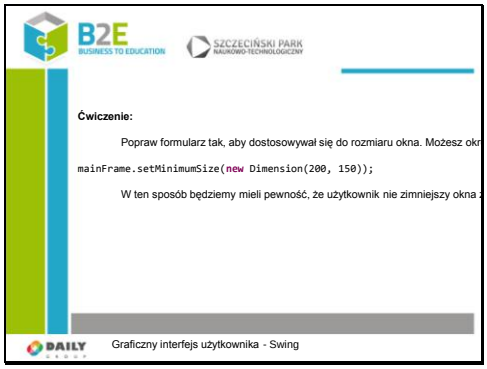


"GridLayout" układa komponenty podobnie jak tabela. Tworząc go należy podać liczbę kolumn i wersów.

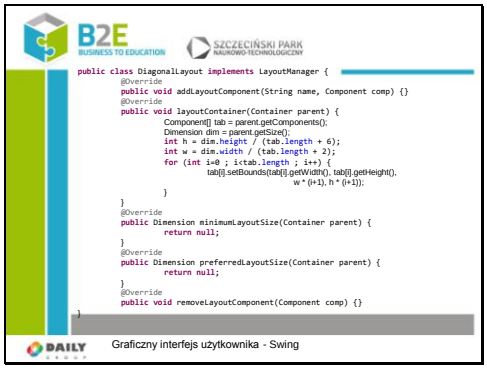
Slajd 24



Slajd 25



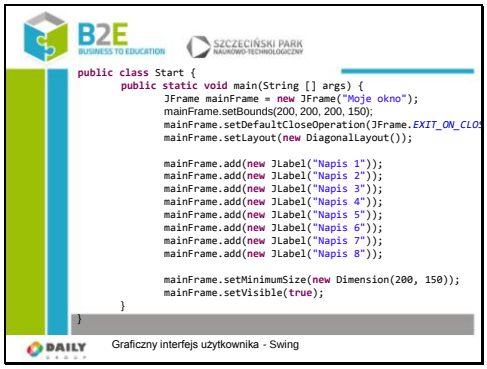
Slajd 26



Swing umożliwia nam napisanie własnego zarządcy rozkładu. Wystarczy zaimplementować "LayoutManager". Zaimplemenujmy przynajmniej metodę "layoutContainer". Odpowiada ona za ponowne rozmieszczenie komponentów po zmianie rozmiaru okna.

Powyższy przykład układa elementy wzdłuż przekątnej okna.

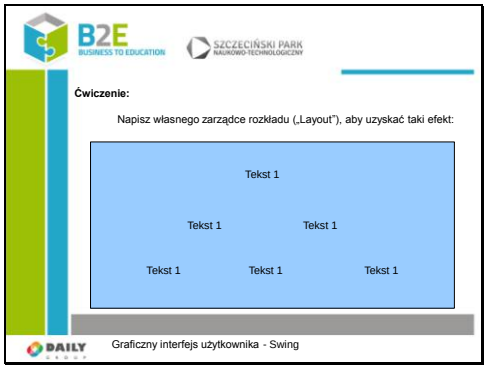
Slajd 27



Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 28

Slajd 29



8.9.3 Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie będą potrafili stworzyć proste okno ze skonfigurowanymi komponentami. Obsłużą zmiany rozmiaru okna.

8.10 Lekcja 10 - Przechwytywanie zdarzeń, tworzenie plików wykonywalnych

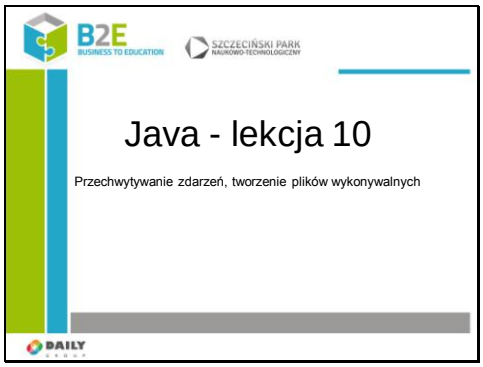
8.10.1 Cel lekcji

Celem lekcji jest pokazanie jak prawidłowo obsługiwać zdarzenia takie jak: ruchy kursora, starowania klawiatura. Omówiona zostanie budowa plików wykonywalnych jar.

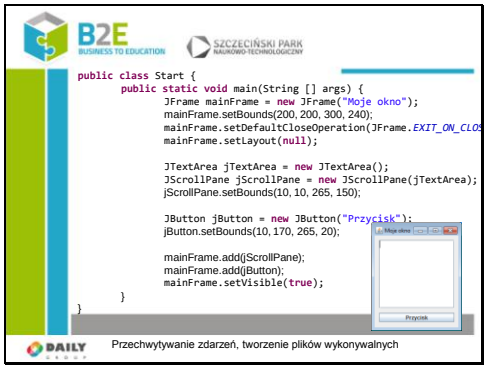
Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

8.10.2 Treść - slajdy z opisem

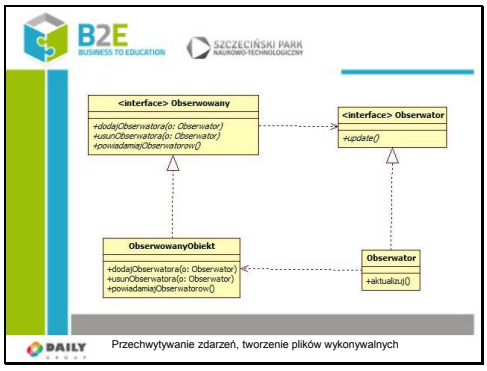
Slajd 1



Slajd 2



Slajd 3



W programowaniu obiektowym obiekty posiadają pewien stan, tj. zbiór aktualnych wartości pól obiektu, który w wyniku wykonywania na nich operacji może ulegać zmianie. Od bieżącego stanu mogą być zależne inne obiekty, dlatego musi istnieć możliwość ich powiadomienia o jego zmianie tak, aby mogły one się do niej dostosować. Możemy także żądać, aby inne obiekty były powiadamiane o tym, że inny obiekt próbuje wykonać konkretną czynność, np. ponownie nawiązywać utracone połączenie z bazą danych. Pragniemy zaimplementować ogólny mechanizm, który umożliwi nam osiągnięcie tych celów.

We wzorcu obserwator wyróżniamy dwa podstawowe typy obiektów:

obserwowany (ang. observable, subject) - obiekt, o którym chcemy uzyskiwać informacje,
obserwator (ang. observer, listener) - obiekty oczekujące na powiadomienie o zmianie stanu obiektu obserwowanego.

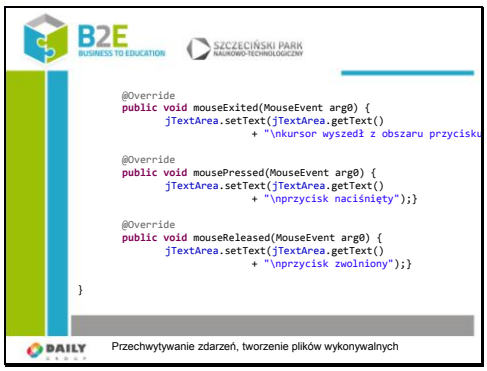
Kiedy stan obiektu obserwowanego się zmienia, wywołuje on metodę "powiadomObserwatorow()", która wysyła powiadomienia do wszystkich zarejestrowanych obserwatorów.

Obserwator jest stosowany w aplikacjach z graficznym interfejsem użytkownika.

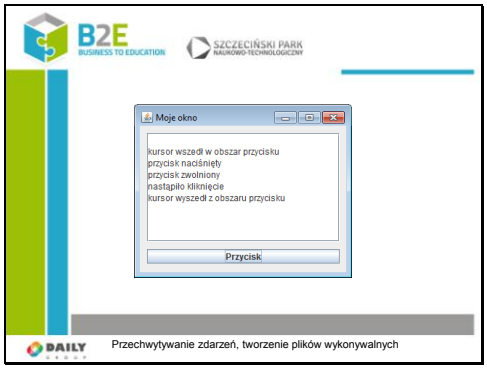
Slajd 4

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

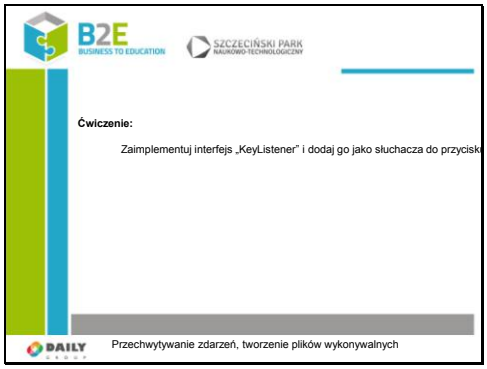
Slajd 5



Slajd 6

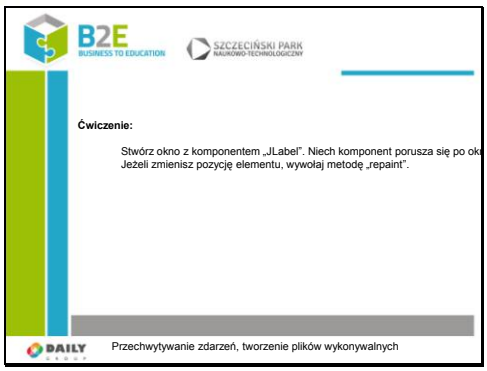


Slajd 7



KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 8



Slajd 9

Do przykładu z początku lekcji dodaj klasę wewnętrzną i zapisz zmiany. Dla pewności uruchom program.

Slajd 10

W projekcie znajdują się trzy pliki źródłowe:

- MyKeyboardListener.java,
- MyListener.java,
- Start.java.

Wejdź do folderu workspace, następnie do folderu z tym projektem. W środku znajdują się dwa foldery "src" i "bin".

Sprawdź ich zawartość. W folderze "src" znajdują się pliki źródłowe i są trzy.

Dlaczego w folderze są cztery pliki?

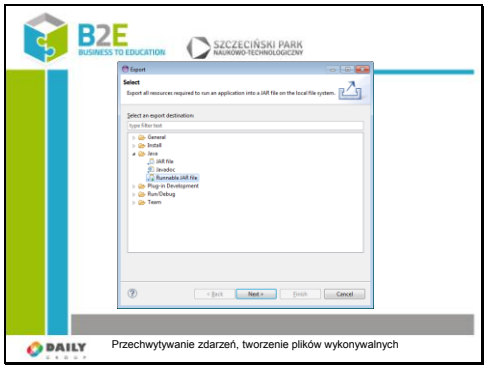
Na jednej z pierwszych lekcji powiedziane było, że w jednym pliku znajdować się może tylko jedna klasa.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

W plikach źródłowych mogą być klasy wewnętrzne, ale po procesie kompilacji każda klasa stanowi osobny plik.

W klasie "Start" jest klasa wewnętrzna "MyInnerClass". Po skompilowaniu jest osobnym plikiem. Po nazwie można łatwo rozpoznać, z której klasy została wyodrębniona.

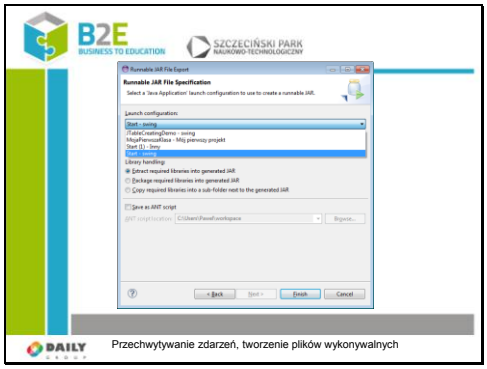
Slajd 11



W środowisku Eclipse kliknij prawym przyciskiem na projekt i użyj opcji eksport.

Wybież opcję "Java/Runnable JAR file".

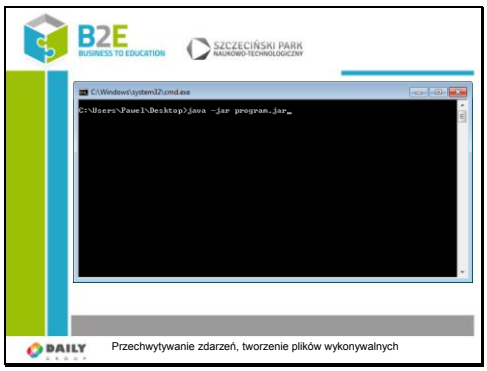
Slajd 12



Wybież plik, który służy do uruchamiania aplikacji. Następnie podaj nazwę i lokalizację pliku.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego

Slajd 13



Uruchom konsolę i wywołaj polecenie tak, jak na slajdzie.

Powinno nastąpić prawidłowe uruchomienie programu.

Zmień rozszerzenie pliku z "jar" na "zip" i rozpakuj go.

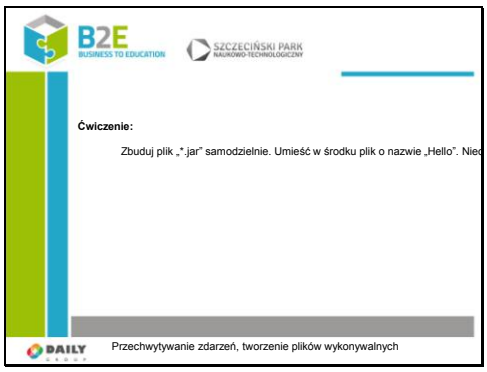
Zauważ, że jest to zwykłe archiwum zip. Znajdują się w nim wszystkie pliki bajtkodu, należące do danego projektu.

W folderze "META-INF" znajduje się plik manifestu. Zobacz jego zawartość.

Manifest-Version: 1.0
Class-Path: .
Main-Class: Start

W ostatnim wersie jest nazwa klasy, z której wywoływany jest program.

Slajd 14



8.10.3 Opis założonych osiągnięć ucznia

Po tej lekcji uczniowie będą umieli obsłużyć zdarzenia systemowe związane z kursorem i klawiaturą. Będą rozumieć jak wygląda tworzenie plików wykonywalnych.

Człowiek - najlepsza inwestycja
KAPITAŁ LUDZKI NARODOWA STRATEGIA SPÓJNOŚCI
UNIA EUROPEJSKA EUROPEJSKI FUNDUSZ SPOŁECZNY
Projekt współfinansowany przez Unię Europejską w ramach Europejskiego Funduszu Społecznego