

KARTA PRZEDMIOTU

1. Informacje ogólne

Nazwa przedmiotu i kod (wg planu studiów):	Projektowanie obiektowe interfejsów bazodanowych D1.2
Nazwa przedmiotu (j. ang.):	Designing object-oriented database interfaces
Kierunek studiów:	Informatyka
Specjalność/specjalizacja:	Technologie internetowe i bazy danych
Poziom kształcenia:	studia I stopnia
Profil kształcenia:	praktyczny (P)
Forma studiów:	studia stacjonarne / studia niestacjonarne
Obszar kształcenia:	nauki techniczne (wg wykazu)
Dziedzina:	nauki techniczne (wg wykazu)
Dyscyplina nauki:	(wg wykazu)
Koordynator przedmiotu:	dr hab. Jan Bazan

2. Ogólna charakterystyka przedmiotu

Przynależność do modułu:	specjalnościowego
Status przedmiotu:	obowiązkowy
Język wykładowy:	polski
Rok studiów, semestr:	II, 4 oraz III, 5
Forma i wymiar zajęć według planu studiów:	stacjonarne - wykład 30h, ćw. Projektowe 30h, ćw. laboratoryjne 30h niestacjonarne - wykład 15h, ćw. projektowe 15h, laboratoryjne 15h
Interesariusze i instytucje partnerskie (nieobowiązkowe)	
Wymagania wstępne / Przedmioty wprowadzające:	Podstawowa wiedza w zakresie programowania. Programowanie I, Programowanie II.

3. Bilans punktów ECTS

Całkowita liczba punktów ECTS (wg planu studiów; 1 punkt = 25-30 godzin pracy studenta, w tym praca na zajęciach i poza zajęciami):	6 (A + B)		
		stacjonarne	Niestacjonarne
A. Liczba godzin wymagających bezpośredniego udziału nauczyciela z podziałem na typy zajęć oraz całkowita liczba punktów ECTS osiągniętych na tych zajęciach	obecność na wykładach	30	15
	obecność na ćwiczeniach lab.	30	15
	ćwiczenia projektowe	30	15
	udział w konsultacjach	5	10
	udział w egzaminie	2	2
W sumie:		97	57
	ECTS	3,5	2,0
B. Poszczególne typy zadań do samokształcenia studenta (niewymagających bezpośredniego udziału nauczyciela) wraz z planowaną średnią liczbą godzin na każde i sumaryczną liczbą ECTS	przygotowanie do zajęć lab. (zad. domowe)	20	30
	przygotowanie do kolokw. zal. przy komp.	15	25
	praca nad projektem	20	25
	przygotowanie do egzaminu	15	20
	w sumie:	70	100
	ECTS	2,5	4,0
C. Liczba godzin praktycznych/laboratoryjnych w ramach przedmiotu oraz związana z tym liczba punktów ECTS	obecność na ćwiczeniach lab.	30	15
	ćwiczenia projektowe	30	15
	praca praktyczna samodzielna (zadania domowe, przygotowanie do kolokwium przy komputerze oraz opracowanie projektu)	55	80
	w sumie:	115	110
		ECTS	4

4. Opis przedmiotu

Cel przedmiotu: Celem przedmiotu jest nabycie przez studentów potrzebnej wiedzy i umiejętności w zakresie programowania graficznych interfejsów użytkownika w obiektowym, wieloplatformowym środowisku Java.
Metody dydaktyczne: wykład - pokaz, laboratorium - zadania problemowe, metoda projektu
Treści kształcenia (w rozbiciu na formę zajęć (jeśli są różne formy) i najlepiej w punktach): Wykłady: 1. Motywacja nauki programowania w języku Java (cechy języka Java, ranking TIOBE). 2. Podstawowe pojęcia dotyczące tworzenia oprogramowania (algorytm, program, programowanie, programista, oprogramowanie) 3. Programowanie orientowane obiektowo (OOP). 4. Geneza języka Java. 5. Zasady użytkowania Java SE Development Kit. 6. Typowe środowiska programistyczne do programowania w języku Java.

7. Rodzaje programów pisanych w języku Java (aplikacje, applety, servlety, midlety, skrypty, facelety) oraz tworzenie dokletów.
8. Ogólna postać aplikacji w języku Java.
9. Identyfikatory, typy, zmienne, instrukcja przypisania, typy proste i klasowe, operatory, wyrażenia, domniemana i jawna konwersja, kod ASCII, klasa String, operacje wejścia-wyjścia i komentarze.
10. Przepływ sterowania programem i sposoby jego modyfikacji.
11. Obiekty, klasy, pola i metody.
12. Pola i metody statyczne.
13. Klasy opakowujące.
14. Przeciążanie nazw metod.
15. Konstruktory i ich przeciążanie.
16. Pakiety (wykorzystanie gotowych i tworzenie nowych).
17. Tablice.
18. Trzy filary programowania OOP (hermetyzacja, dziedziczenie i polimorfizm).
19. Przeciążanie a przesłanianie metod.
20. Modyfikatory dostępu do pól i metod.
21. Zaskakująca konsekwencja polimorfizmu.
22. Wyjątki (definiowanie, rzucanie, chwytanie, wyjątki standardowe).
23. Strumienie i pliki.
24. Zasady tworzenia metod rekurencyjnych w języku Java.
25. Rysowanie w Javie.
26. Wprowadzenie do tworzenia graficznych interfejsów użytkownika z użyciem biblioteki Swing i JavaFX.
27. Tworzenie appletów
28. Odśmiecacz pamięci.
29. Archiwa Jar (wykorzystanie i tworzenie).
30. Typy parametryzowane.
31. Adnotacje.
32. Standardowe kolekcje w języku Java oraz operacje na nich, w tym sortowanie.
33. Programowanie aplikacji łączących się z bazą danych.
34. Tworzenie aplikacji internetowych działających po stronie serwera (tzw. cienki klient).
35. Tworzenie aplikacji działających w architekturze klient-server (tzw. grupy klient).
36. Programowanie aplikacji wielowątkowych.
37. Obsługa XML, w tym parsery XML.
38. Tworzenie dokletów.
39. Sposoby dystrybuowania programów napisanych w języku Java.
40. Dekompilacja i obfuskacja (zaciemnianie kodu).

Ćwiczenia (laboratoryjne):

1. Wprowadzenie użytkownika do użytkownika środowiska IDE dla Java SE (edytor kodu, struktura programu, automatyczne formatowanie kodu, kompilacja i uruchamianie programów).
2. Prosty przykład programu z operacjami wejścia-wyjścia.
3. Korzystanie z JDK bez IDE.
4. Programy demonstrujące operatory i ich priorytety oraz konwersję domniemaną oraz jawną.
5. Programy z obiektami klasy String.

6. Programy z instrukcjami warunkowymi i pętlami (while, do while, for, instrukcje break i continue).
7. Przykłady programów składających się z dwóch lub trzech klas.
8. Przykłady programów demonstrujących metody statyczne oraz klasę biblioteczną Math, klasy opakujące i ich możliwości, przeciążanie nazw metod w klasach i konstruktory klas, gotowe pakiety oraz definiowanie własnych pakietów.
9. Przykłady programów wykorzystujących obsługę wyjątków.
10. Kilka programów z tablicami, także z tablicą dwuwymiarową.
11. Przykłady programów wykorzystujących polimorfizm.
12. Programy zapisujące i odczytujące pliki.
13. Kilka przykładów programów rekurencyjnych napisanych w języku Java.
14. Po jednym zadaniu programistycznym na każdą z 5 standardowych struktur danych, czyli kolekcji ArrayList, LinkedList, HashSet, TreeSet i HashMap.
15. Kilka programów używających GUI w Swingu.
16. Kilka programów używających GUI w JavaFX.
17. Dwa programy tworzące rysunki (koniecznie z wyświetlaniem pliku graficznego i z animacją).
18. Przykład apletu i powiązanie tworzenia apletów z biblioteką Swing.
19. Dwa zadania programistyczne na parsowanie plików XML
20. Przykład aplikacji komunikującej się z bazą danych z użyciem sterownika JDBC.
21. Przykład programu tworzonego z użyciem technologii JSP.
22. Przykład na programowanie aplikacji klient-serwer z użyciem gniazd.
23. Przykłady programów wielowątkowych.
24. Tworzenie wykonywalnych jar-ów i instalatorów za pomocą generatora.
25. Instalowanie aplikacji webowych na serwerach.
26. Zaliczenie projektów.

5. Efekty kształcenia i sposoby weryfikacji

Efekty kształcenia		
Efekt przedmiotu <i>(kod przedmiotu + kod efektu kształcenia)</i>	Student, który zaliczył przedmiot (spełnił minimum wymagań)	Efekt kierunkowy
Wiedza:		
D1.2_K_W01	1. Ma podstawową wiedzę na temat języka Java i tworzenia w nim oprogramowania.	K_W06, K_W08, K_W14, K_W18,
D1.2_K_W02	2. Ma podstawową wiedzę dotyczącą programowania GUI w języku Java w aplikacjach działających bezpośrednio pod systemem operacyjnym lub opartych na wykorzystaniu stron WWW (po stronie klienta lub po stronie serwera).	K_W06, K_W07, K_W08, K_W14, K_W16
D1.2_K_W03	3. Ma podstawową wiedzę w zakresie tworzenia programów w języku Java mających dostęp do relacyjnej bazy danych.	K_W06, K_W08
	Umiejętności	

D1.2_U01	1. Potrafi tworzyć proste aplikacje przy wykorzystaniu środowiska JDK.	K_U01, K_U03, K_U10, K_U17, K_U19
D1.2_U02	2. Potrafi tworzyć aplikacje wyposażone w GUI w języku Java działające bezpośrednio pod systemem operacyjnym lub oparte na wykorzystaniu stron WWW i uruchamiane za pomocą przeglądarki internetowej (po stronie klienta lub po stronie serwera).	K_U01, K_U03, K_U04, K_U12, K_U13, K_U17, K_U18, K_U19, K_U22, K_U30, K_U31
D1.2_U03	3. Potrafi tworzyć aplikacje w języku Java mające dostęp do relacyjnych baz danych.	K_U01, K_U03, K_U16, K_U17, K_U20

Sposoby weryfikacji efektów kształcenia:

Lp.	Efekt przedmiotu	Sposób weryfikacji	Ocena formująca	Ocena końcowa
1	D1.2_W01	egzamin pisemny	ocena z egzaminu w zakresie tego efektu	średnia z ocen formujących, sprawdzających nabytą wiedzę i umiejętności
2	D1.2_W02	egzamin pisemny	ocena z egzaminu w zakresie tego efektu	
3	D1.2_W03	egzamin pisemny	ocena z egzaminu w zakresie tego efektu	
4	D1.2_U01	kolokwium przy komputerze	ocena z kolokwium w zakresie tego efektu	
5	D1.2_U02	zaliczenie projektu	ocena z projektu w zakresie tego efektu	
6	D1.2_U03	zaliczenie projektu	ocena z projektu w zakresie tego efektu	

Kryteria oceny

w zakresie wiedzy		Efekt kształcenia
Na ocenę 3,0	Student ma podstawową wiedzę na temat języka Java i tworzenia w nim oprogramowania pozyskaną na wykładzie (patrz wyżej w treściach kształcenia). Jednak nie zawsze zna znaczenie rzadziej używanych słów kluczowych (np. abstract, catch, interface, finally, implements, instanceof, native, protected, synchronized, super i inne) oraz nie zna bardziej złożonych standardowych klas bibliotecznych języka Java (np. LinkedList, TreeSet, File, StringBuffer, NumberFormatException, HashMap, HashSet, GZIPOutputStream, GZIPInputStream i inne). Ponadto, nie zawsze potrafi wyjaśnić niektóre bardziej zaawansowane pojęcia dotyczące tworzenia oprogramowania w języku Java (np. hermetyzacja, dziedziczenie, polimorfizm, klasa opakowująca, przeciążanie i przesłanianie metod, odśmieczacz pamięci, kod	D1.2_K_W01

	bajtowy, sarta i wiedza jak zmienić jej rozmiar, stos i wiedza jak zmienić jego rozmiar itd.).	
Na ocenę 5,0	Student ma podstawową wiedzę na temat języka Java i tworzenia w nim oprogramowania pozyskaną na wykładzie (patrz wyżej). Ponadto, zna znaczenie rzadziej używanych słów kluczowych (patrz wyżej), zna bardziej złożone standardowe klasy biblioteczne języka Java (patrz wyżej) i potrafi wyjaśnić bardziej zaawansowane pojęcia dotyczących tworzenia oprogramowania w języku Java (patrz wyżej).	
Na ocenę 3,0	Student ma podstawową wiedzę dotyczącą programowania GUI w języku Java w aplikacjach działających bezpośrednio pod systemem operacyjnym lub opartych na wykorzystaniu stron WWW (po stronie klienta lub po stronie serwera). Jednak nie zawsze zna zaawansowane klasy służące do tworzenia bardziej złożonych komponentów GUI (np. klasy do tworzenia gotowych tabel, zakładek, list, edytorów tekstowych, okien wyboru kolorów, okien wczytywania lub zapisu plików), klas do tworzenia bardziej zaawansowanych menedżerów ułożenia komponentów, klas do tworzenia menu, niestandardowej obsługi myszki i klawiatury, klas do obsługi pasków postępu i przewijania, poleceń (metod) dotyczących zaawansowanego wykreślenia elementów itd. Ponadto, nie zawsze zna bardziej zaawansowane mechanizmy tworzenia systemów webowych (np. dyrektywy, obiekty deklarowane niejawnie, strony diagnostyczne, mechanizmy śledzenia sesji, obsługa cookies itd.)	D1.2_W02
Na ocenę 5,0	Student ma podstawową wiedzę dotyczącą programowania GUI w języku Java w aplikacjach działających bezpośrednio pod systemem operacyjnym lub opartych na wykorzystaniu stron WWW (po stronie klienta lub po stronie serwera). Ponadto, zna zaawansowane klasy służące do tworzenia bardziej złożonych komponentów GUI (patrz wyżej), klasy do tworzenia bardziej zaawansowanych menadżerów ułożenia komponentów, klasy do tworzenia menu, klasy do niestandardowej obsługi myszki i klawiatury, klasy do obsługi pasków postępu i przewijania, polecenia zaawansowanego wykreślenia elementów. Zna także zaawansowane mechanizmy tworzenia systemów webowych (patrz wyżej).	
Na ocenę 3,0	Student ma podstawową wiedzę w zakresie tworzenia programów w języku Java mających dostęp do relacyjnej bazy danych. Jednak nie zna bardziej zaawansowanych mechanizmów dostępu do bazy danych z poziomu języka Java (np. meta dane i mechanizmy ich wykorzystania, pule połączeń itd.)	
Na ocenę 5,0	Student ma podstawową wiedzę w zakresie tworzenia programów w języku Java mających dostęp do relacyjnej bazy danych. Zna także bardziej zaawansowane mechanizmy dostępu do bazy danych z poziomu języka Java (patrz wyżej)	D1.2_W03

w zakresie umiejętności		Efekt kształcenia
Na ocenę 3,0	Student potrafi tworzyć proste aplikacje przy wykorzystaniu środowiska JDK, przy czym oprócz podstawowych elementów języka nie potrafi wykorzystać w swoim programie wskazanego elementu spośród następującej listy elementów: strumienie, standardowe klasy biblioteczne (nie liczymy tutaj String, bo to jest podstawa), dziedziczenie i polimorfizm, wyjątki, implementacja obliczeń w oparciu o prostą teorię matematyczną).	D1.2_U01
Na ocenę 5,0	Student potrafi tworzyć proste aplikacje przy wykorzystaniu środowiska JDK, wykorzystując elementy ze wskazanej wyżej listy.	
Na ocenę 3,0	Student potrafi tworzyć aplikacje wyposażone w GUI w języku Java działające bezpośrednio pod systemem operacyjnym lub uruchamiane przez JVM przeglądarki internetowej. Jednak nie wykorzystuje zaawansowanych klasy służących do tworzenia bardziej złożonych komponentów GUI (np. klasy do tworzenia gotowych tabel, zakładek, list, edytorów tekstowych, okien wyboru kolorów, okien wczytywania lub zapisu plików), klas do tworzenia bardziej zaawansowanych menadżerów ułożenia komponentów, klas do tworzenia menu, niestandardowej obsługi myszki i klawiatury, klas do obsługi pasków postępu i przewijania, poleceń (metod) dotyczących zaawansowanego wykreślenia elementów itd.	D1.2_U02
Na ocenę 5,0	Student potrafi tworzyć aplikacje wyposażone w GUI w języku Java działające bezpośrednio pod systemem operacyjnym lub uruchamiane przez JVM przeglądarki internetowej, wykorzystując zaawansowane klasy służące do tworzenia bardziej złożonych komponentów GUI.	
Na ocenę 3,0	Student potrafi tworzyć aplikacje serwerowe wyposażone w GUI w języku Java oparte na wykorzystaniu dynamicznych stron WWW. Jednak nie wykorzystuje klasy bardziej zaawansowanych mechanizmów stosowanych do tego celu (np. dyrektywy, obiekty dekladowane niejawnie, znaczniki akcji, strony diagnostyczne, mechanizmy śledzenia sesji, cookies itd.)	D1.2_U03
Na ocenę 5,0	Student potrafi tworzyć aplikacje serwerowe wyposażone w GUI w języku Java oparte na wykorzystaniu dynamicznych stron WWW, wykorzystując zaawansowane klasy bardziej zaawansowanych mechanizmów (patrz wyżej).	
Na ocenę 3,0	Student potrafi tworzyć aplikacje w języku Java mające dostęp do relacyjnych baz danych. Jednak nie wykorzystuje bardziej zaawansowanych mechanizmów dostępu do bazy danych z poziomu języka Java (np. meta dane i mechanizmy ich wykorzystania, pule połączeń itd.)	D1.2_U04
Na ocenę 5,0	Student potrafi tworzyć aplikacje w języku Java mające dostęp do relacyjnych baz danych, wykorzystując bardziej zaawansowane mechanizmy dostępu do bazy danych z poziomu języka Java.	

Student, który osiągnął zakładany poziom wiedzy, posiadał wymagane umiejętności, cechuje się określonymi kompetencjami społecznymi, które są zdefiniowane w efektach kształcenia dla przedmiotu, zalicza przedmiot.

Student, który nie osiągnął zakładanych efektów kształcenia, nie zalicza przedmiotu.

Kryteria oceny końcowej:

Zaliczenie przedmiotu następuje na podstawie zaliczenia wszystkich efektów weryfikowanych przez planowane w danym okresie metody weryfikacji. Przy czym zakłada się, że każda metoda weryfikacji dostarcza osobne oceny dla każdego z weryfikowanych przez nią efektów kształcenia. Jeśli dany efekt jest weryfikowany przez więcej niż jedną metodę, to ocena weryfikująca osiągnięcie tego efektu jest obliczana jako średnia arytmetyczna ocen uzyskanych w poszczególnych metodach weryfikowania tego efektu.

Student otrzymuje ocenę **niedostateczny**, gdy metody weryfikacji wykażą, iż przynajmniej jeden efekt nie został osiągnięty (średnia ocena dla tego efektu jest niższa niż 3.0);

Student otrzymuje ocenę **dostateczny**, gdy przeciętnie każdy z efektów zostanie osiągnięty na poziomie co najmniej 3.0, ale chociaż jeden z efektów został osiągnięty na poziomie mniejszym od 3.75;

Student otrzymuje ocenę **dobry**, gdy przeciętnie każdy z efektów zostanie osiągnięty na poziomie co najmniej 3.75, ale chociaż jeden z efektów został osiągnięty na poziomie mniejszym od 4.75;

Student otrzymuje ocenę **bardzo dobry**, gdy przeciętnie każdy z efektów zostanie osiągnięty na poziomie co najmniej 4.75;

Zalecana literatura :

Podstawowa:

1. Wykłady: <http://fenix.univ.rzeszow.pl/bazan/>
2. Schildt, H.: Java. Przewodnik dla początkujących, wydanie VI, Gliwice: Helion (2015) (dostępna w bibliotece PWSZ).
3. Rychlicki, W.: Programowanie w języku Java. Zbiór zadań z (p)odpowiedziami, Gliwice: Helion (2012) (dostępna w bibliotece PWSZ)
4. Horstman, C., S., Cornell, G.: Java, Podstawy, wydanie IX, Gliwice: Helion (2013) (dostępna w bibliotece PWSZ).

Uzupełniająca:

1. Schildt, H.: Java. Kompendium programisty, wydanie IX, Gliwice: Helion (2015) (dostępna w bibliotece PWSZ).
2. Horstman, C., S., Cornell, G.: Java, Techniki zaawansowane, wydanie IX, Gliwice: Helion (2013) (dostępna w bibliotece PWSZ).
3. Eckel, B.: *Thinking in Java – edycja polska*, Wydanie IV, Gliwice: Helion (2006) (dostępna w bibliotece PWSZ).
4. Witryny internetowe na temat: Swing, JavaFX, JSP i JDBC.

Informacje dodatkowe:

Dodatkowe obowiązki prowadzącego wraz z szacowaną całkowitą liczbą godzin:
Przygotowanie wykładów, opracowanie ćwiczeń laboratoryjnych, przygotowanie tematów projektu - 10 godzin
Konsultacje – 5 godzin
Poprawa prac projektowych – 5 godzin
Przygotowanie i poprawa egzaminu – 5 godzin
W sumie: 25 godzin