

Programowanie obiektowe i zaawansowane metody programowania - opis przedmiotu

Informacje ogólne	
Nazwa przedmiotu	Programowanie obiektowe i zaawansowane metody programowania
Kod przedmiotu	13.2-WI-GeoTSP-Pr.obiekt.-S18
Wydział	Wydział Budownictwa, Architektury i Inżynierii Środowiska
Kierunek	Geoinformatyka i techniki satelitarne
Profil	ogólnoakademicki
Rodzaj studiów	pierwszego stopnia z tyt. inżyniera
Semestr rozpoczęcia	semestr zimowy 2019/2020

Informacje o przedmiocie	
Semestr	2
Liczba punktów ECTS do zdobycia	4
Typ przedmiotu	obowiązkowy
Język nauczania	polski
Sylabus opracował	dr inż. Tomasz Gratkowski

Formy zajęć					
Forma zajęć	Liczba godzin w semestrze (stacjonarne)	Liczba godzin w tygodniu (stacjonarne)	Liczba godzin w semestrze (niestacjonarne)	Liczba godzin w tygodniu (niestacjonarne)	Forma zaliczenia
Wykład	15	1	9	0,6	Egzamin
Laboratorium	45	3	45	3	Zaliczenie na ocenę

Cel przedmiotu

Prezentowanie podstawowych pojęć związanych z programowaniem obiektowym (klasa, obiekt, abstrakcja, hermetyzacja, polimorfizm, dziedziczenie) oraz ich realizacją w wykorzystaniu języka programowania Java. Ukształtowanie potrzeby projektowania programów z zastosowaniem wzorców projektowych. Zapoznanie z nowoczesnymi narzędziami programistycznymi wspierającymi wytwarzanie oprogramowania obiektowego na platformie Java.

Wymagania wstępne

Brak

Zakres tematyczny

1. Wprowadzenie - Omówienie środowiska Java Development Kit oraz zintegrowanych środowisk deweloperskich IDE dla platformy Java. Podstawy kompilacji i uruchamiania programów na platformie Java.
2. Algorytmy - Definicje algorytmu, cechy algorytmu, metody zapisu algorytmów, złożoność obliczeniowa
3. Reprezentacja liczb, zmienne - Typy danych, zmienne proste i referencyjne, literały, operatory, tablice, instrukcje sterujące, zasięg widoczności zmiennych, funkcje, właściwości zmiennych.
4. Myślenie w sposób obiektowy - Klasy i instancje, właściwości klas i metod. Zasady konstrukcji obiektów i poznanie mechanizmu czyszczenia pamięci.
5. Dziedziczenie - Dziedziczenie, polimorfizm i hermetyzacja. Projektowanie złożonych typów obiektowych przy użyciu kompozycji i dziedziczenia.
6. Tajniki abstrakcji - Tworzenie interfejsów programistycznych z wykorzystaniem klas abstrakcyjnych i interfejsów. Rozszerzanie interfejsów. Klasy wewnętrzne i klasy statycznie zagnieżdżone.
7. Kolekcje - Przechowywanie złożonych informacji w aplikacji (Listy, kolejki, zbiory, mapy i algorytmy).
8. Obsługa wyjątków & Korzystanie z innych bibliotek - Zabezpieczenie kodu i przechwytywanie wyjątków. Operacje wyjścia/wejścia - co są strumienie.
9. Tworzenie graficznego interfejsu użytkownika

Metody kształcenia

Wykład - wykład konwencjonalny z wykorzystaniem wideoprojektora.

Laboratorium - zajęcia praktyczne w laboratorium komputerowym.

Efekty uczenia się i metody weryfikacji osiągnięcia efektów uczenia się

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Ma wiedzę na temat procesu wytwarzania oprogramowania w oparciu o technologię Java	K_W04 K_U06	bieżąca kontrola na zajęciach	Laboratorium
Zna zasady projektowania i programowania obiektowego	K_W04	egzamin - ustny, opisowy, testowy i inne	Wykład
Zna definicje podstawowych paradygmatów programowania	K_W04	egzamin - ustny, opisowy, testowy i inne	Wykład

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
Potrafi skompilować i uruchomić samodzielnie napisaną aplikację w języku Java	K_U06	bieżąca kontrola na zajęciach	Laboratorium

Warunki zaliczenia

Wykład - egzamin w formie pisemnej i/lub w formie testu

Laboratorium – ocena końcowa stanowi sumę ważoną ocen uzyskanych ze sprawdzianów kontrolnych weryfikujących przygotowanie merytoryczne do ćwiczeń oraz za realizację poszczególnych ćwiczeń laboratoryjnych

Ocena końcowa = 50 % oceny zaliczenia z formy zajęć wykład + 50 % oceny zaliczenia z formy zajęć laboratorium.

Literatura podstawowa

Sierra K., Bates B.: Java. Rusz głową! Wydanie II, Helion, 2010

Horstmann, C.S., Cornell, G.: Java., Podstawy, Wydanie dziewiąte, Gliwice, Helion, 2013

Horstmann, C.S., Cornell, G., Java. Techniki zaawansowane, Wydanie dziewiąte, Gliwice, Helion, 2013

Wróblewski, M.: Algorytmy, struktury danych i techniki programowania. Wydanie V, Helion, 2015

Cormen T. H., Leiserson C. E., Rivest R. L., Stein C., Wprowadzenie do algorytmów, Wydawnictwo Naukowe PWN, 2018

Literatura uzupełniająca

Martin, R.C.: Czysty kod. Podręcznik dobrego programisty, Helion, 2010

Eckel, B., Thinking in Java, Wydanie IV, Warszawa, Helion, 2006

Lis, M., Praktyczny kurs Java, Wydanie II, Gliwice, Helion, 2004

Coldwind G., Zrozumieć programowanie, Wydawnictwo Naukowe PWN, 2018

Uwagi

Zmodyfikowane przez dr inż. Tomasz Gratkowski (ostatnia modyfikacja: 23-04-2019 18:05)

Wygenerowano automatycznie z systemu SylabUZ