

Autor: dr inż. Daniel Paczesny

Algorytmy i programowanie 2 **Algorithms and programming 2**

Poziom kształcenia: I stopień

Forma i tryb prowadzenia przedmiotu: stacjonarna

Kierunek studiów: Cyberbezpieczeństwo

Specjalność: -

Grupa przedmiotów:-

Poziom przedmiotu: podstawowy

Status przedmiotu: obowiązkowy

Język przedmiotu: polski

Semestr nominalny (tylko dla przedmiotów obowiązkowych): 2

Minimalny numer semestru: 2

Wymagania wstępne, zalecane przedmioty poprzedzające: ALPO1

Limit liczby studentów: 60

Powód zgłoszenia przedmiotu: program studiów na nowym kierunku Cyberbezpieczeństwo

Cel przedmiotu:

Przedmiot ten jest kontynuacją przedmiotu Algorytmy i programowanie 1 i przedstawia wybrane zagadnienia z zakresu implementacji zagadnień algorytmicznych i rozwiązań realizowanych w języku Java. Przedmiot ma na celu poszerzenie wiedzy oraz umiejętności w zakresie stosowania wybranych struktur danych (kolejka priorytetowa, kopiec, drzewa binarne, tablice haszujące) oraz rozwiązań algorytmicznych (sortowanie Shella, sortowanie przez wstawianie, wyszukiwanie sekwencyjne, binarne czy operacje na grafach). Przedstawione zostaną dodatkowe możliwości języka Java. Student po ukończeniu przedmiotu zdobędzie wiedzę i umiejętnościami w zakresie: implementacji złożonych struktur danych i algorytmów w postaci kodu programu, tworzenia oprogramowania zrozumiałego dla innych członków zespołu programistycznego, tworzenia oprogramowania bazując na wzorcach projektowych oraz testowania wytworzonego oprogramowania. Zostaną przedstawione zagadnienia z zakresu tworzenia bezpiecznych programów oraz budowania programów wielowątkowych.

Treść kształcenia:

WYKŁADY:

1. Wprowadzenie (2 godz.)

Informacja o przedmiocie. Regulamin przedmiotu. Przypomnienie najważniejszych informacji z poprzedniego semestru. Ciąg dalszy dobrych praktyk programistycznych. Wprowadzenie do języka UML.

Wydziałowa Komisja Akredytacji Przedmiotów (WKAP)

2. Wejście-wyjście, strumień (2 godz.)

Wejście-wyjście. Narzędzia do odczytu danych z pliku i zapis do pliku. Pliki binarne. Buforowanie. Język XML.

3. Analizowanie algorytmów (2 godz.)

Metodologia, studium przypadku.

4. Algorytmy sortowania (4 godz.)

Sortowanie Shella, sortowanie przez scalanie, zastosowania. Struktury danych: kolejka priorytetowa, kopiec.

5. Algorytmy wyszukiwania (4 godz.)

Wyszukiwanie sekwencyjne, binarne, zastosowania. Wyszukiwanie podłańcuchów znaków. Struktury danych: tablica symboli, drzewa binarne, tablice haszujące.

6. Internacjonalizacja (1 godzina.)

Formaty liczb, waluty, data i czas, formatowanie komunikatów, wczytywanie i wyświetlanie tekstów.

7. Algorytmy grafowe (4 godz.)

Graf nieskierowany, reprezentacja danych, przeszukiwanie, minimalne drzewo rozpinające, poszukiwanie najkrótszej ścieżki.

8. Bezpieczeństwo w Java (2 godz.)

Ładowanie klas, menadżer bezpieczeństwa, uwierzytelnianie użytkowników, wstęp do szyfrowania.

9. Testowanie programów (1 godz.)

Obsługa błędów za pomocą wyjątków. Tworzenie własnych wyjątków. Przechwytywanie wyjątku, instrukcja *finally*. Adnotacje. Programowanie sterowane testami, refaktoryzacja.

10. Wybrane wzorce projektowe (4 godz.)

Fasada, mediator, singleton, monostate, fabryka, wizytator, stan.

11. Współbieżność. (2 godz.)

Klasa Thread. Współdzielenie zasobów.

12. Sprawdzenie efektów uczenia się (2 godz.)

Pierwsze kolokwium po zrealizowaniu trzeciego tematu (1 godz.). Drugie kolokwium kończące przedmiot (1 godz.)

LABORATORIA:

1. Organizacja laboratoriów. Ćwiczenia w zakresie wejścia-wyjścia programu, pliki tekstowe – xml, pliki binarne.
2. Implementacja własnej struktury danych np. lista, kolejka, itp.

Wydziałowa Komisja Akredytacji Przedmiotów (WKAP)

3. Implementacja wybranej struktury danych, implementacja i testowanie algorytmów sortujących.
4. Implementacja wybranej struktury danych, implementacja i testowanie algorytmów wyszukiwujących
5. Implementacja wybranej struktury danych, implementacja i testowanie algorytmów grafowych.

PROJEKT:

Celem wykonania projektu jest samodzielne opracowanie i wykonanie programu komputerowego w języku Java.

Projekt powinien obejmować:

1. Rozwiązanie postawionego zagadnienia algorytmicznego w postaci kodu programu w tym opracowanie lub zastosowanie odpowiednich struktur danych,
2. Przygotowanie odpowiedniej strategii testowania oprogramowania i przeprowadzenie testów,
3. Implementacje wzorców projektowych i uzasadnienie wyboru,
4. Tworzenie bezpiecznego oprogramowania.

Treść kształcenia - streszczenie w jęz. angielskim:

The aim of this module is to teach the students advanced programming techniques using Java in order to support its use on other modules. This module also aims to teach the common data structures and algorithms. Having successfully completed this module, student will be able to demonstrate knowledge and understanding of: construct advanced Java applications, construct multi-threaded Java applications, use persistent storage for Java applications, perform testing, validation and verification on Java programs, construct Java application for sorting, searching and graphs algorithms.

Egzamin: nie

Literatura i oprogramowanie:

- Noel Kalicharan (2014) - Advanced Topics in Java: Core Concepts in Data Structures
- Cay S. Horstmann (2017) - Core Java, Volume II - Advanced Features
- Robert C. Martin (2015) - Clean Code: A Handbook of Agile Software Craftsmanship
- Robert C. Martin (2017) - The Clean Coder: A Code of Conduct for Professional Programmers
- Robert C. Martin - Agile Software Development, Principles, Patterns, and Practices
- Robert Sedgewick, Kevin Wayne (2012) - Algorithms
- Java Language and Virtual Machine Specifications: <https://docs.oracle.com/javase/specs/>
- The Java Tutorials: <https://docs.oracle.com/javase/tutorial/index.html>

Wymiar godzinowy zajęć:

W	C	L	P
30	–	10	12

Przewidywane formy kształcenia i organizacja przedmiotu

Realizacja przedmiotu obejmuje następujące formy zajęć:

Wydziałowa Komisja Akredytacji Przedmiotów (WKAP)

- wykład prowadzony w wymiarze 2 godz. tygodniowo; w wybranych zagadnieniach przewidziana jest aktywizacja studentów na wykładzie,
- zajęcia laboratoryjne w wymiarze 2 godz. tygodniowo (5 spotkań); w ramach tych zajęć student, korzystając z oprogramowania i sprzętu komputerowego, będąc pod opieką prowadzącego zajęcia, będzie realizował wskazane zadania dotyczące tworzenia własnych struktur danych, użycia dostępnych struktur danych oraz implementacji i testowania wybranych algorytmów.
- projekt realizowany w drugiej części semestru, jako samodzielne rozwiązanie postawionego zadania. W trakcie spotkań projektowych będą omawiane na przykładach wybrane problemy oraz ich rozwiązania związane z tematami i zakresem realizowanych projektów.

Sprawdzanie założonych efektów kształcenia realizowane jest przez:

- ocenę wiedzy i umiejętności związanych z realizacją zadań laboratoryjnych – ocena z wybranych ćwiczeń laboratoryjnych oraz wykonania projektu;
- ocenę wiedzy wykazanej na dwóch kolokwium pisemnych.

Wymiar w jednostkach ECTS: 4 pkt.

Liczba godzin pracy studenta związanych z osiągnięciem efektów kształcenia (opis):

1. liczba godzin kontaktowych – **54 godz.**, w tym
 - obecność na wykładach: **30 godz.**,
 - obecność na zajęciach laboratoryjnych: **10 godz.**,
 - obecność na zajęciach projektowych: **12 godz.**,
 - udział w konsultacjach związanych z realizacją przedmiotu: **2 godz.**
2. praca własna studenta – **54 godz.**, w tym
 - analiza literatury i materiałów wykładowych związana z przygotowaniem do kolejnych wykładów, wskazanie przykładów do wykonania: **18 godz.**
 - przygotowanie się do ćwiczeń laboratoryjnych: **10 godz.**
 - wykonanie zagadnień projektowych: **16 godz.**
 - przygotowanie do kolokwium: **10 godz.**

Łączny nakład pracy studenta wynosi 108 godz., co odpowiada 4pkt. ECTS.

Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich: 2 pkt. ECTS, co odpowiada 54 godz. kontaktowym.

Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym: 2 pkt. ECTS, co odpowiada $22 + 26 = 48$ godz. realizacji ćwiczeń i projektu

Wydziałowa Komisja Akredytacji Przedmiotów (WKAP)

Efekty kształcenia / uczenia się i formy ich weryfikacji:

efekty kształcenia/uczenia się	forma zajęć/ technika kształcenia	Sposób weryfikacji (oceny)*	odniesienie do efektów uczenia się dla programu
student, który zaliczył przedmiot:			
WIEDZA			
w1: ma wiedzę dotyczącą tworzenia programów z wykorzystaniem technik zorientowanych obiektowo	wykład + laboratorium + projekt	kolokwium, laboratorium, projekt	W05, W08
w2: ma wiedzę dotyczącą metod dokumentowania tworzonych programów	wykład + laboratorium + projekt	kolokwium, laboratorium, projekt	W05, W08, W10
w3: ma wiedzę dotyczącą dobrych praktyk programistycznych	wykład + laboratorium + projekt	kolokwium, laboratorium, projekt	W05, W08, W10
w4: ma wiedzę dotyczącą metod dekompozycji zadania i realizacji w postaci kodu programu z użyciem języka programowania zorientowanego obiektowo	wykład + laboratorium + projekt	kolokwium, laboratorium, projekt	W05, W08
w5: ma wiedzę dotyczącą konstruowania algorytmów, zna zagadnienia dotyczące złożoności algorytmu	wykład + laboratorium + projekt	kolokwium, laboratorium, projekt	W05, W08
w6: ma wiedzę dotyczącą konstrukcji struktur danych i ich zastosowania	wykład + laboratorium + projekt	kolokwium, laboratorium, projekt	W05, W08
w7: ma wiedzę dotyczącą środowisk i procesów związanych z wytwarzaniem programów	wykład + laboratorium + projekt	kolokwium, laboratorium, projekt	W05, W08, W10
UMIĘTNOŚCI			
u1: potrafi dokonać krytycznej analizy dostępnych źródeł w zakresie realizowanego zadania	wykład + laboratorium + projekt	laboratorium, projekt	U08, U09, U13
u2: potrafi przygotować koncepcję rozwiązania postawionego problemu algorytmicznego	wykład + laboratorium + projekt	laboratorium, projekt	U08, U09, U10
u3: potrafi zaprojektować (wybrać lub opracować) odpowiednie struktury danych do postawionego problemu algorytmicznego	wykład + laboratorium + projekt	laboratorium, projekt	U08, U09
u4: potrafi napisać kod programu wykorzystując możliwości języka obiektowego	wykład + laboratorium + projekt	laboratorium, projekt	U08, U09
u5: potrafi przygotować zestaw testów oraz potrafi przetestować poprawność wytworzonego oprogramowania	wykład + laboratorium + projekt	laboratorium, projekt	U08, U09, U11
u6: potrafi dokumentować wytworzone oprogramowanie oraz potrafi zaprezentować swoje osiągnięcia	wykład + laboratorium + projekt	laboratorium, projekt	U10
KOMPETENCJE SPOŁECZNE			
ks1: rozumie potrzebę stałego aktualizowania i wzbogacania posiadanej wiedzy.	wykład + laboratorium + projekt	kolokwium, laboratorium, projekt	KS01
ks2: ma świadomość konieczności komunikowania się z otoczeniem, także pozazawodowym, w sposób zrozumiały dla odbiorcy	wykład + laboratorium + projekt	laboratorium, projekt	KS05

Uwagi:

Data i podpis autora (kierownika zespołu autorskiego):