

**Autor:** dr inż. Daniel Paczesny

## **Algorytmy i programowanie 1** **Algorithms and programming 1**

**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):** 1

**Minimalny numer semestru:** 1

**Wymagania wstępne, zalecane przedmioty poprzedzające:** -

**Limit liczby studentów:** 60

**Powód zgłoszenia przedmiotu:** program studiów na nowym kierunku Cyberbezpieczeństwo

### **Cel przedmiotu:**

Przedmiot ten ma na celu nauczanie podstawowych struktur danych i algorytmów, które stanowią fundament nowoczesnej inżynierii oprogramowania, podstawowych zasad, które towarzyszą algorytmom i strukturom danych oraz podstaw programowania z użyciem języka zorientowanego obiektowo, jakim jest Java. Student po zakończeniu przedmiotu będzie mógł się wykazać wiedzą z zakresu zrozumienia złożoności obliczeniowej, zastosowania języka obiektowego do konstruowania struktur danych, konstruowania prostych algorytmów obliczeniowych. Student zdobędzie umiejętności w zakresie implementacji prostych struktur danych w postaci kodu programu, zastosowania odpowiedniego algorytmu do rozwiązania postawionego problemu i zaimplementowania go w postaci kodu programu, posługiwania się zintegrowanym środowiskiem programistycznym oraz sprawnego posługiwania się obiektowym językiem programowania. Student zapozna się również z podstawami konstruowania programu komputerowego w zakresie zastosowania zmiennych, wyrażeń, obiektów, klas, dziedziczenia, konstrukcji programistycznych oraz typów danych.

### **Treść kształcenia:**

#### **WYKŁADY:**

##### **1. Wprowadzenie (2 godz.)**

Informacje o przedmiocie. Regulamin przedmiotu. Podstawowe pojęcia: obiekt, program, algorytm. Konstrukcja pierwszego programu komputerowego. Uruchomienie programu.

## Wydziałowa Komisja Akredytacji Przedmiotów (WKAP)

Debugger. Wirtualna Maszyna Java. Biblioteki. Zintegrowane środowisko uruchomieniowe. System Linux.

### **2. Podstawy algorytmizacji (2 godz.)**

Wstęp do projektowania algorytmu. Dekompozycja problemu. Sprawność i złożoność czasowa algorytmu. Cechy algorytmów. Projektowanie algorytmów. Dane wejściowe i efekt działania algorytmu. Konstruowanie i testowanie prostych algorytmów. Sortowanie bąbelkowe. Wprowadzenie do struktur danych. Komentowanie kodu programu.

### **3. Wstęp do obiektów i podstawowy model programowania (7 godz.)**

Tworzenie obiektów. Przechowywanie danych. Proste typy danych, pola. Tworzenie własnych typów danych. Metody i ich przeciążanie. Konstruktor, inicjalizacja i sprzątanie. Operatory. Sterowanie przebiegiem wykonania programu. Zasięg obiektów. Kontrola dostępu: public, private, protected. Mechanizmy sterowania wejściem do programu i wyjściem.

### **4. Własności obiektów (4 godz.)**

Pakiety. Kompozycja. Dziedziczenie. Rzutowanie. Polimorfizm. Interfejsy.

### **5. Typy danych, kolekcje obiektów, podstawowe struktury danych (4 godz.)**

Obiekt class. Tablice. Łańcuchy znaków. Kontenery. Abstrakcja danych: wielozbiory i lista.

### **6. Podstawowe algorytmy sortowania. (6 godz.)**

Analiza algorytmów. Sortowanie przez wybieranie, sortowanie przez wstawianie, sortowanie szybkie. Porównanie algorytmów sortujących.

### **7. Testowanie programów (2 godz.)**

Projektowanie oprogramowania i testowanie. Obsługa błędów za pomocą wyjątków. Standardowe wyjątki Java. Słowo kluczowe this. Testy jednostkowe.

### **8. Dobre praktyki programistyczne (1 godz.)**

Podsumowanie zagadnień związanych z: nazewnictwem, tworzeniem metod, dokumentowanie i formatowanie kodu programu.

### **9. Sprawdzenie efektów uczenia się (2 godz.)**

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

## LABORATORIA:

1. Organizacja laboratoriów. System operacyjny i polecenia systemu operacyjnego linux. Przesyłanie plików. Standard komunikacyjny ssh. Wirtualna maszyna Java. Uruchomienie pierwszego programu.
2. Zintegrowane środowisko programistycznego. Uruchamianie programu. Debugger. Komentowanie kodu programu.

## Wydziałowa Komisja Akredytacji Przedmiotów (WKAP)

3. Wersjonowanie oprogramowania, umieszczanie kodów programów w repozytorium, praca grupowa. Zapoznanie się z wybranymi bibliotekami. Dokumentacja i sprawozdania.
4. Testowanie prostych typów danych, operatory, sterowanie przebiegiem wykonania programu.
5. Konstruowanie metod. Testowanie różnych wersji sortowania bąbelkowego.
6. Tworzenie własnych typów danych. Konstruktor, inicjalizacja i sprzątanie. Komunikacja obiektów ze światem zewnętrznym.
7. Ćwiczenia z zakresu zasięgu obiektu, kontroli dostępu: private, public, protected. Zagadnienia wejścia-wyjścia.
8. Testowanie cech obiektów: interfejsy, polimorfizm, dziedziczenie, kompozycja.
9. Podstawowe struktury danych. Konstruowanie list i kolejek. Konstrukcje wbudowane języka.
10. Implementacja wybranych algorytmów sortowania. Porównanie algorytmów sortowania i ocena złożoności i szybkości działania.
11. Testowanie programów. Obsługa sytuacji wyjątkowych. Testy jednostkowe.
12. Praca nad zadaniem projektowym - rozdanie i omówienie tematów dla grup projektowych. Omówienie dokumentacji.
13. Praca nad zadaniem projektowym - ocena przyjętych założeń, ocena szkieletu programu. Przypomnienie dobrych praktyk programistycznych.
14. Praca nad zadaniem projektowym - sprawdzenie aktualnego stanu zadania. Konsultacje w grupach.
15. Praca nad zadaniem projektowym - zakończenie i odebranie zadania.

### **Treść kształcenia - streszczenie w jęz. angielskim:**

- This module aims to introduce students to the principles of programming using an object oriented approach, and to provides them with the programming skills necessary to continue the study of cybersecurity. Java is used as the introductory language. Additionally module teaches the basic data structures and algorithms which underpins modern software engineering. Without these algorithms most software would be hopelessly slow to the point of unusability. Having successfully completed this module, student will be able to demonstrate knowledge and understanding of: simple object oriented terminology, including classes, objects, inheritance and methods; basic programming constructs including sequence, selection and iteration, the use of identifiers, variables and expressions, and a range of data types; good programming style; knowledge of common data structures and algorithms; understanding of time complexity and understanding of how to code data structures using object oriented methods.

### **Egzamin: nie**

### **Literatura i oprogramowanie:**

- Bruce Eckel (2006) - Thinking in Java,
- Cay S. Horstmann (2016) - Core Java Volume I - Fundamentals
- Robert Sedgewick, Kevin Wayne (2012) - Algorithms
- Java Language and Virtual Machine Specifications: <https://docs.oracle.com/javase/specs/>

## Wydziałowa Komisja Akredytacji Przedmiotów (WKAP)

- The Java Tutorials: <https://docs.oracle.com/javase/tutorial/index.html>

**Wymiar godzinowy zajęć:**

W	C	L	P
30	–	30	–

### Przewidywane formy kształcenia i organizacja przedmiotu

Realizacja przedmiotu obejmuje następujące formy zajęć:

- wykład prowadzony w wymiarze 2 godz. tygodniowo; w wybranych zagadnieniach przewidziana jest aktywizacja studentów na wykładzie,
- zajęcia laboratoryjne w wymiarze 2 godz. tygodniowo; w ramach tych zajęć student, korzystając z oprogramowania i sprzętu komputerowego, będąc pod opieką prowadzącego zajęcia, będzie realizował wskazane ćwiczenia dotyczące podstaw języka programowania Java oraz podstaw implementacji algorytmów; studenci podzieleni na grupy zadaniowe będą zobowiązani do wspólnego rozwiązywania zadania.

Sprawdzanie założonych efektów kształcenia realizowane jest przez:

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

**Wymiar w jednostkach ECTS: 5 pkt.**

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

1. liczba godzin kontaktowych – **64 godz.**, w tym
  - obecność na wykładach: **30 godz.**,
  - obecność na zajęciach laboratoryjnych: **30 godz.**,
  - udział w konsultacjach związanych z realizacją przedmiotu: **4 godz.**
2. praca własna studenta – **76 godz.**, w tym
  - analiza literatury i materiałów wykładowych związana z przygotowaniem do kolejnych wykładów, wskazanie przykładów do wykonania: **20 godz.**
  - wykonanie ćwiczeń przygotowawczych do laboratorium: **30 godz.**
  - wykonanie zadania projektowego: **16 godz.**
  - przygotowanie do kolokwium: **10 godz.**

**Łączny nakład pracy studenta wynosi 140 godz., co odpowiada 5pkt. ECTS.**

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:** 2.25 pkt. ECTS, co odpowiada 64 godz. kontaktowym.

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:** 3 pkt. ECTS, co odpowiada  $60 + 16 = 76$  godz. realizacji ćwiczeń

# 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:			
<b>WIEDZA</b>			
<b>w1:</b> ma wiedzę dotyczącą fundamentalnych pojęć z zakresu programowania obiektowego w tym: klas, obiektów, dziedziczenia, metod i funkcji, używania bibliotek-pakietów	wykład + laboratorium	kolokwium, laboratorium	W05
<b>w2:</b> ma podstawową z zakresu wybranych konstrukcji programistycznych, kolejności wykonywania kodu programu, stosowania instrukcji sterujących, operatorów i zmiennych.	wykład + laboratorium	kolokwium, laboratorium	W05
<b>w3:</b> ma podstawową wiedzę z zakresu stosowania dobrych praktyk programistycznych	wykład + laboratorium	kolokwium, laboratorium	W05
<b>w4:</b> ma podstawową wiedzę z zakresu fundamentalnych pojęć dotyczących algorytmów komputerowych	wykład + laboratorium	kolokwium, laboratorium	W05
<b>w5:</b> ma podstawową wiedzę dotyczącą konstruowania algorytmów, zna zagadnienia dotyczące złożoności i sprawności algorytmu	wykład + laboratorium	kolokwium, laboratorium	W05
<b>w6:</b> ma podstawową wiedzę dotyczącą konstrukcji podstawowych struktur danych i ich zastosowania	wykład + laboratorium	kolokwium, laboratorium	W05
<b>w7:</b> ma podstawową wiedzę z zakresu wykorzystania systemów komputerowych do archiwizowania wytworzonego oprogramowania, pracy wspólnej w zespole.	wykład + laboratorium	kolokwium, laboratorium	W05
<b>UMIEJĘTNOŚCI</b>			
<b>u1:</b> potrafi przygotować środowisko pracy w tym zintegrowane środowisko programistyczne niezbędne do uruchamiania programów napisanych w języku Java	wykład + laboratorium	kolokwium, laboratorium	U04, U09
<b>u2:</b> potrafi wykorzystać podstawowe konstrukcje języka w tym rozwiązania obiektowe	wykład + laboratorium	kolokwium, laboratorium	U08, U09
<b>u3:</b> potrafi napisać prosty program, użyć debuggera, uruchomić i przetestować zbudowany program	wykład + laboratorium	kolokwium, laboratorium	U08, U09
<b>u4:</b> potrafi napisać kod programu wykorzystujący proste struktury danych	wykład + laboratorium	kolokwium, laboratorium	U08, U09
<b>u5:</b> potrafi samodzielnie rozwiązać proste zagadnienia programistyczne implementując proste algorytmy obliczeniowe	wykład + laboratorium	kolokwium, laboratorium	U03, U09
<b>u6:</b> potrafi przygotować prostą dokumentację przedstawiającą rozwiązanie zadanego problemu	wykład + laboratorium	kolokwium, laboratorium	U10
<b>u7:</b> potrafi wyszukiwać niezbędne informacje w zasobach literaturowych	wykład + laboratorium	kolokwium, laboratorium	U13
<b>u08:</b> potrafi pracować indywidualnie i w zespole	wykład + laboratorium	kolokwium, laboratorium	U09
<b>KOMPETENCJE SPOŁECZNE</b>			
<b>ks1:</b> rozumie potrzebę stałego aktualizowania i wzbogacania posiadanej wiedzy.	wykład + laboratorium	kolokwium, laboratorium	KS01
<b>ks2:</b> ma świadomość konieczności komunikowania się z otoczeniem, także pozazawodowym, w sposób zrozumiały dla odbiorcy	wykład + laboratorium	laboratorium	KS05

**Uwagi:**

**Data i podpis autora (kierownika zespołu autorskiego):**