

ECTS – Arkusz przedmiotu

| | | | | | | | |
|---|--|----------------|--------------------------|---|------------------|-----------|-------------|
| Kod | PIP_20E_33IZ_s | | Nazwa przedmiotu | Programowanie i algorytmy optymalizacji kombinatorycznej Programming and combinatorial optimization algorithms | | | |
| Prowadzący przedmiot | Waldemar Kaczmarczyk | | | | | | |
| Osoby prowadzące zajęcia | Waldemar Kaczmarczyk | | | | | | |
| Klasa przedmiotu | O | | Rodzaj przedmiotu | E | | | |
| Wydział | ZARZĄDZANIA | | | | | | |
| Kierunek/Specjalność | Zarządzanie i Inżynieria Produkcji | | | Inżynieria Zarządzania | | | |
| Rodzaj studiów | s | | Stopień studiów | 2 | Semestr | 3 | |
| Rodzaje zajęć | Suma | Wykłady | Ćwiczenia | Laboratoria | Seminaria | DL | ECTS |
| Liczba godzin | 30 | 15 | 15 | - | - | - | 1 |
| WWW | | | | | | | |
| Uwagi | | | | | | | |
| Cel przedmiotu - zdobyte umiejętności | | | | | | | |
| Przygotowanie studentów do samodzielnej analizy i rozwiązywania kombinatorycznych problemów optymalizacyjnych z zakresu logistyki i zarządzania produkcją. Studenci mają opanować język, Python, wybrane zagadnienia inżynierii oprogramowania i podstawy heurystycznych algorytmów optymalizacji kombinatorycznej. | | | | | | | |
| Streszczenie przedmiotu | | | | | | | |
| Najpierw przedstawiony zostanie język Python, umożliwiający szybkie opanowanie programowanie z uwzględnieniem metodologii obiektowej. Następnie przy jego pomocy przedstawiona zostanie: reprezentacja podstawowych struktur danych optymalizacji kombinatorycznej, m.in. zbiorów, grafów, kolejek priorytetowych, sekwencji, metody tworzenie i wykorzystanie własnych typów danych przy pomocy metodologii obiektowej, a wreszcie algorytmy heurystyczne, konstrukcyjne i ulepszające, oraz ich implementacja dla podstawowych zadań planowania produkcji i logistyki. Przedstawione zostaną przykłady wykorzystania wzorców projektowych. | | | | | | | |
| Warunki uczestnictwa w przedmiocie | Uczestnictwo w wykładach, ćwiczeniach. | | | | | | |
| Forma zaliczenia przedmiotu | Realizacja samodzielnych projektów i kolokwium zaliczeniowe. | | | | | | |
| Zasada wystawiania oceny końcowej | Średnia ocena projektów i kolokwium zaliczeniowego. | | | | | | |
| Program wykładów | | | | | | | |
| <ol style="list-style-type: none"> 1. Programowanie proceduralne w Pythonie: podstawowe struktury danych, polecenia i funkcje. 2. Programowanie obiektowe w Pythonie: klasy i obiekty, własne typy danych, polimorfizm. 3. Ogólne zasady dobrego programowania proceduralnego i obiektowego. 4. Implementacja wybranych struktur danych przydatnych w optymalizacji kombinatorycznej, m.in.: listy skierowane, kolejki, grafy, zapis kolejności i przydziału. 5. Podstawy algorytmów heurystycznych, konstrukcyjnych i ulepszających. 6. Obiektowa implementacja algorytmów heurystyk lokalnego przeszukiwania. 7. Przykłady wykorzystania wzorców projektowych. | | | | | | | |
| Program pozostałych zajęć (ćwiczenia, laboratoria, projekty, seminaria) | | | | | | | |
| <ol style="list-style-type: none"> 1. Programowanie proceduralne w Pythonie: podstawowe struktury danych, polecenia i funkcje. 2. Programowanie obiektowe w Pythonie: klasy i obiekty, własne typy danych, polimorfizm. 3. Ogólne zasady dobrego programowania proceduralnego i obiektowego. 4. Implementacja wybranych struktur danych przydatnych w optymalizacji kombinatorycznej, m.in.: listy skierowane, kolejki, grafy, zapis kolejności i przydziału. 5. Podstawy algorytmów heurystycznych, konstrukcyjnych i ulepszających. | | | | | | | |

6. Obiektowa implementacja algorytmów heurystyk lokalnego przeszukiwania.
7. Przykłady wykorzystania wzorców projektowych.

Bibliografia

1. Guido van Rossum, Fred L. Drake, Jr., *Przewodnik po języku Python*, Python-Labs, część oryginalnej dokumentacji Pythona, darmowa książka dostępna w Internecie.
2. Mark Lutz, *Python, Wprowadzenie*, Helion, 2009.
3. Allen B. Downey, Jerey Elkner, Chris Meyers, *How to Think Like a Computer Scientist Learning with Python*, darmowa książka dostępna w Internecie.
4. Steve McConnell, *Code Complete: A Practical Handbook of Software Construction*, Microsoft Press, Redmond 1993.
5. Braian W. Kernighan, Rob Pike, *Lekcja programowania*, z serii *Inżynieria oprogramowania*, WNT, 2002, (ang. *The Practice of Programming*, Addison-Wesley, Reading, Massachusetts 1999).
6. Alan Shalloway, James R. Trott, *Projektowanie zorientowane obiektowo: wzorce projektowe*, Helion, Gliwice, 2001.
7. Maciej M. Sysło, Narsingh Deo, Janusz S. Kowalik, *Algorytmy optymalizacji dyskretnej*, Wydawnictwo Naukowe PWN, 1999.
8. Zbigniew Michalewicz, David B. Fogel, *Jak to rozwiązać, czyli nowoczesna heurystyka*, WNT, 2006.