

POLITECHNIKA KRAKOWSKA IM. TADEUSZA KOŚCIUSZKI

KARTA PRZEDMIOTU

obowiązuje studentów rozpoczynających studia w roku akademickim 2023/2024

Wydział Informatyki i Telekomunikacji

Kierunek studiów: Informatyka

Profil: Ogólnoakademicki

Forma studiów: stacjonarne

Kod kierunku: I

Stopień studiów: I

Specjalności: Brak specjalności

1 INFORMACJE O PRZEDMIOCIE

| | |
|---|-------------------------------------|
| NAZWA PRZEDMIOTU | Języki i paradygmaty programowania |
| NAZWA PRZEDMIOTU W JĘZYKU ANGIELSKIM | Programming languages and paradigms |
| KOD PRZEDMIOTU | WiIT I oIS C2 23/24 |
| KATEGORIA PRZEDMIOTU | Przedmioty kierunkowe |
| LICZBA PUNKTÓW ECTS | 11.00 |
| SEMESTRY | 1 2 |

2 RODZAJ ZAJĘĆ, LICZBA GODZIN W PLANIE STUDIÓW

| SEMESTR | WYKŁAD | ĆWICZENIA | LABORATORIUM | LABORATORIUM KOMPUTERO- WE | SEMINARIUM | PROJEKT |
|---------|--------|-----------|--------------|----------------------------------|------------|---------|
| 1 | 30 | 0 | 0 | 30 | 0 | 0 |
| 2 | 30 | 0 | 0 | 30 | 0 | 15 |

3 CELE PRZEDMIOTU

Cel 1 Poznanie podstawowych paradygmatów programowania, w tym paradygmatu proceduralnego oraz obiektowego. Osiągnięcia umiejętności w ocenie przydatności paradygmatów do rozwiązywania różnego typu proble-

mów; projektowania, implementacji, testowania i debugowania prostych programów proceduralnych i obiektowych.

Cel 2 Poznanie składni języków C/C++ jako przykładów języka strukturalnego oraz obiektowego.

Cel 3 Poznanie podstaw proceduralnego podejścia do programowania - tworzenia funkcji oraz procedur, które potrafią obsłużyć dowolne typy danych. Zamykanie danych w oddzielnych plikach, udostępnianie działań nad danymi przez funkcje - elementy stylu obiektowego w języku C.

Cel 4 Poznanie podstaw obiektowego podejścia do programowania - enkapsulacja, dziedziczenie i polimorfizm.

4 WYMAGANIA WSTĘPNE W ZAKRESIE WIEDZY, UMIEJĘTNOŚCI I INNYCH KOMPETENCJI

1 Brak wymagań wstępnych

5 EFEKTY KSZTAŁCENIA

EK1 Wiedza Student rozumie pojęcie paradygmat programowania. Ma wiedzę ogólną w zakresie języków i paradygmatów programowania, programowania obiektowego. Potrafi wybrać paradygmat właściwy dla problemu, który rozwiązuje oraz zna środowiska (języki) umożliwiające implementację rozwiązania problemu.

EK2 Wiedza Student zna składnię C/C++, rozumie na czym polegają proceduralne oraz obiektowe podejścia do programowania. Ma szczegółową wiedzę nt. algorytmiki, projektowania i programowania obiektowego.

EK3 Umiejętności Student potrafi czytać ze zrozumieniem programy napisane w C/C++, potrafi napisać i uruchomić własny program w C/C++, który rozwiązuje postawiony przed nim problem.

EK4 Umiejętności Umie stosować styl obiektowy przy tworzeniu prostych programów w języku C przy użyciu struktur i unii, potrafi rozdzielić obsługę tablic, kolejek, stosów od obsługi danych. Umie stosować techniki języków obiektowych.

6 TREŚCI PROGRAMOWE

| PROJEKT | | |
|---------|--|------------------|
| LP | TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BŁOKÓW TEMATYCZNYCH | LICZBA GODZIN |
| P1 | Tworzenie projektu w środowisku MS VS. Dodawanie plików źródłowych. Struktura projektu VS. | 2 |
| P2 | Klasa obsługi błędów i komunikatów na podstawie wyjątków C++. Awaryjne przerywanie aplikacji. | 3 |
| P3 | Obsługa wejścia-wyjścia. Tworzenie katalogu i ustalenie ścieżki. Zapis danych do pliku binarnego i odczyt. Wyprowadzenie danych do pliku w formacie EXCEL. | 4 |
| P4 | Obsługa interfejsu. | 2 |
| P5 | Klasa danych. Przeciążenie operatorów >, =, , ==. | 2 |
| P6 | Kontener tablicy dynamicznej, zrealizowany w postaci klasy-szablonu. | 2 |

| WYKŁAD | | |
|--------|--|------------------|
| LP | TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH | LICZBA GODZIN |
| W1 | Pojęcie języka strukturalnego, struktura programu napisanego w języku C, pliki źródłowe i nagłówkowe. Struktura biblioteki CRT. Pojęcie o IDE. Pojęcie funkcji, funkcji statyczne, strumieni stdin, stdout, znaki specjalne (escape sequence), typy i formaty zmiennych, dyrektywy preprocesora. | 2 |
| W2 | Operatory, wyrażenia, instrukcje, operator przypisania. Biblioteka funkcji matematycznych. Konwersja typów, instrukcja warunkowa, operatory arytmetyczne, relacyjne, logiczne, priorytety operatorów. | 2 |
| W3 | Definicja a deklaracja funkcji, lista argumentów, argumenty formalne i faktyczne, sposoby przekazywania argumentów do funkcji, wartość zwracana przez funkcję. | 2 |
| W4 | Instrukcje pętli for, while, do-while, instrukcja goto. Pojęcie o wskaźnikach. Konwersje typów - jawne i niejawne. | 2 |
| W5 | Stałe i zmienne znakowe. Tablice typu char, wchar_t, int, double. Stałe tekstowe. Tablice dwuwymiarowe. Tablice w argumentach funkcji. | 2 |
| W6 | Wskaźniki do tablic. Dynamiczne alokowanie i zwolnienie pamięci dla tablic jednowymiarowych. Poruszanie się po tablicy za pomocą wskaźnika. Funkcje CRT obsługi tablic. Zmienna enum. | 2 |
| W7 | Statyczne wydzielenie pamięci i inicjalizacja. Dynamiczne alokowanie i zwolnienie pamięci. Pojęcie o fragmentowaniu przestrzeni adresowej i sposoby kompresji heap. Kopiowanie tablic na przekrywających się obszarach pamięci oraz inne operacje nad tablicami. Funkcje CRT obsługi heap. | 2 |
| W8 | Dynamiczne alokowanie i zwolnienie pamięci dla tablic dwuwymiarowych. Inicjalizacja, pojęcie "wskaźnik do wskaźnika". | 2 |
| W9 | Otwieranie i zamykanie pliku. Zapis i odczyt sformatowany. Pliki binarne. Sterowanie wskaźnikiem pozycji pliku. Zapis i odczyt do/z pliku binarnego. Tworzenie i usunięcie katalogu. Struktura pliku z rekordami o różnej długości. | 2 |
| W10 | Struktura programu w C. Zmienne lokalne, globalne, statyczne. Klasy pamięci i zasięg deklaracji (obszar widoczności). | 2 |
| W11 | Pojęcie struktury jako łączenie danych w całość. Obiekty struktur oraz tablice struktur. Dynamiczne alokowanie i zwolnienie pamięci. | 2 |
| W12 | Wektory i macierze. Podstawowe algorytmy algebry liniowej i osobliwości ich implementacji w języku C. Rozwijanie pętli i blokowanie rejestrów. | 2 |
| W13 | Struktury danych: kolejka, stos, lista. Rozdzielenie obsługi kontenera od obsługi danych. Wskaźniki void * oraz wskaźniki do funkcji. Tablice wskaźników do funkcji. | 2 |
| W14 | Bitowe operacje logiczne. Maski bitowe. | 2 |
| W15 | Pojęcia o wektoryzowaniu obliczeń za pomocą technik SSE2, AVX, AVX2, AVX512F. | 2 |

| WYKŁAD | | |
|------------|--|------------------|
| LP | TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH | LICZBA GODZIN |
| W16 | Programowanie obiektowe: inkapsulacja, polimorfizm i dziedziczenie. Zasięg deklaracji i czas trwania obiektów. Przestrzenie nazw. | 2 |
| W17 | Wprowadzenie do klas. Konstruktorzy i destruktory. | 2 |
| W18 | Wprowadzenie w dziedziczenie. Specyfikatory private, protected, public. Funkcje inline. Przypisanie obiektów. Przekazywanie obiektów do funkcji. | 2 |
| W19 | Zwracanie obiektu przez funkcję. Funkcje zaprzyjaźnione. | 2 |
| W20 | Przeciążenie funkcji, konstruktory kopiujące, argumenty domyślne. | 2 |
| W21 | Przeciążenie operatorów. | 2 |
| W22 | Dynamiczne alokowanie pamięci. Operatory new, delete. | 2 |
| W23 | Dziedziczenie. Specyfikatory dostępu. Wielodziedziczenie. Klasy wirtualne. | 2 |
| W24 | Polimorfizm dynamiczny. Funkcje wirtualne, abstrakcyjne, klasy abstrakcyjne. | 2 |
| W25 | Szablony funkcji i klas. | 2 |
| W26 | Identyfikacje typu na etapie wykonania (RTTI) | 2 |
| W27 | Wejście-wyjście w C++. | 4 |
| W28 | Obsługa wyjątków. Statyczne składowe klasy. Specyfikatory const, volatile. Niepolimorficzne rzutowanie typów. | 2 |
| W29 | Wprowadzenie w STL. | 2 |

| LABORATORIUM KOMPUTEROWE | | |
|--------------------------|---|------------------|
| LP | TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH | LICZBA GODZIN |
| K1 | Wprowadzenie danych z klawiatury i wyprowadzenie na monitor. Formatowanie, tworzenie tabeli. Operacje arytmetyczne, priorytet operacji, konwersja typów. Instrukcje if. | 2 |
| K2 | Tablice jednowymiarowe i dwuwymiarowe. Pętle for, while, do - while. | 2 |
| K3 | Obsługa wierszy tekstowych: wprowadzenie danych z monitora, kopiowanie, sklepanie, sortowanie, poszukiwanie kontekstu. | 2 |
| K4 | Tablice o dwóch indeksach przy dynamicznym alokowaniu pamięci. Zmiana rozmiaru wiersza, wymiana wierszy pomiędzy sobą, dodawanie nowego wiersza. | 2 |
| K5 | Tablica obiektów struktur przy dynamicznym alokowaniu pamięci. Obsługa tablicy, obsługa obiektu. | 2 |

| LABORATORIUM KOMPUTEROWE | | |
|--------------------------|--|------------------|
| LP | TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH | LICZBA GODZIN |
| K6 | Wskaźniki do funkcji: program obliczający całkę oznaczoną z gładkiej funkcji jednej zmiennej rzeczywistej. | 2 |
| K7 | Stworzenie systemu zapisu do pliku binarnego tablicy obiektów. Różne rekordy mają różną długość. Odczyt z pliku binarnego rekordów w dowolnej kolejności. | 2 |
| K8 | Kolokwium I. | 2 |
| K9 | Hierarchia obiektów. Obsługa tablicy obiektów Triangle, zawierających obiekt Point (wierzchołek trójkąta). Generalna zasada: kontener (tablica) nie powinna zależeć od typu danych. Technika: stosowanie wskaźników void * oraz wskaźników do funkcji. | 2 |
| K10 | Hierarchia obiektów. Obsługa tablicy obiektów Triangle, zawierających obiekt Point (wierzchołek trójkąta). Teraz Point wymaga dynamicznego alokowania pamięci dla tablicy współrzędnych oraz jej zwolnienia. | 4 |
| K11 | Tworzymy dla poprzedniego zadania menedżer pamięci, odpowiadający za alokowanie, zwolnienie, re-alokowanie w celu przechwytywania wycieku pamięci. | 2 |
| K12 | Przypisanie jednego obiektu do innego. Powstające problemy i sposoby rozwiązania. | 2 |
| K13 | Techniki obsługi macierzy i wektorów. Rozwijanie pętli dla algorytmu obliczenia iloczynu skalarnego dwóch wektorów, instrukcje SSE, AVX, AVX2, AVX512F. Funkcje pomiaru czasu. | 2 |
| K14 | Kolokwium II. | 2 |
| K15 | Wprowadzenie w C++. Pierwsze programy. | 2 |
| K16 | Enkapsulacja. Pojęcia klasy. Dane i metody klasy | 2 |
| K17 | Inicjowanie i niszczenie obiektu. Konstruktorzy i destruktory. | 2 |
| K18 | Hierarchia klas, wprowadzenie w dziedziczenie. Funkcje inline. | 2 |
| K19 | Funkcje zaprzyjaźnione do klasy. | 2 |
| K20 | Operatory new, delete. Dynamiczne alokowanie i zwolnienie pamięci. | 2 |
| K21 | Przekazywanie obiektów do funkcji (przez wartość, przez wskaźnik, przez referencję). Konstruktory kopiujące. | 2 |
| K22 | Przeciążenie operatorów binarnych. Przeciążenie operatora przypisania. | 2 |
| K23 | Funkcje-szablony. Opracowanie własnego algorytmu Find. | 2 |
| K24 | Klasa-szablon my_vect. Tworzenie projektu z wieloma plikami. Klasy obsługi komunikatów na podstawie wyjątków C++ i interfejsu. Klasy danych. | 4 |
| K25 | Praca z plikami tekstowymi i binarnymi. Odczyt i zapis dokumentu. | 2 |

| LABORATORIUM KOMPUTEROWE | | |
|--------------------------|--|------------------|
| LP | TEMATYKA ZAJĘĆ OPIS SZCZEGÓŁOWY BLOKÓW TEMATYCZNYCH | LICZBA GODZIN |
| K26 | Funkcje wirtualne, identyfikacje typu RTTI. | 2 |
| K27 | Kolokwium 1 | 2 |
| K28 | Kolokwium 2 | 2 |

7 NARZĘDZIA DYDAKTYCZNE

N1 Wykłady

N2 Ćwiczenia laboratoryjne

N3 Konsultacje

N4 Zadanie projektowe

N5 MS TEAMS

N6 strona <https://torus.uck.pk.edu.pl/fialko.sergiy/> (Sergiy Fialko)

8 OBCIĄŻENIE PRACĄ STUDENTA

| FORMA AKTYWNOŚCI | ŚREDNIA LICZBA GODZIN NA ZREALIZOWANIE AKTYWNOŚCI |
|--|---|
| Godziny kontaktowe z nauczycielem akademickim, w tym: | |
| Godziny wynikające z planu studiów | 135 |
| Konsultacje przedmiotowe | 10 |
| Egzaminy i zaliczenia w sesji | 5 |
| Godziny bez udziału nauczyciela akademickiego wynikające z nakładu pracy studenta, w tym: | |
| Przygotowanie się do zajęć, w tym studiowanie zalecanej literatury | 45 |
| Opracowanie wyników | 45 |
| Przygotowanie raportu, projektu, prezentacji, dyskusji | 35 |
| SUMARYCZNA LICZBA GODZIN DLA PRZEDMIOTU WYNIKAJĄCA Z CAŁEGO NAKŁADU PRACY STUDENTA | 275 |
| SUMARYCZNA LICZBA PUNKTÓW ECTS DLA PRZEDMIOTU | 11.00 |

9 SPOSOBY OCENY

OCENA FORMUJĄCA

F1 Odpowiedź ustna

F2 Projekt indywidualny

F3 Kolokwium

F4 Egzamin

OCENA PODSUMOWUJĄCA

P1 Średnia ważona ocen formujących

WARUNKI ZALICZENIA PRZEDMIOTU

W1 Zaliczenie projektu indywidualnego.

W2 Napisanie kolokwium na pozytywną ocenę.

W3 Zaliczenie Egzaminu na pozytywną ocenę.

W4 Spełnienie warunku obecności na zajęciach

KRYTERIA OCENY

| EFEKT KSZTAŁCENIA 1 | |
|---------------------|--|
| NA OCENĘ 2.0 | Student nie spełnia warunków określonych dla oceny 3.0 |
| NA OCENĘ 3.0 | Student zna zasady proceduralnego/obiekowego programowania i potrafi stworzyć prosty program w języku C/C++. |
| NA OCENĘ 3.5 | Student zna zasady proceduralnego/obiekowego programowania i potrafi stworzyć złożony program w języku C/C++, składający się z wielu plików. |
| NA OCENĘ 4.0 | Student zna zasady proceduralnego/obiekowego programowania i potrafi stworzyć złożony program w języku C/C++, obsługiwać We/Wy do/z pliku binarnego, opracować w programie osobne systemy obsługi błędów i komunikatów. |
| NA OCENĘ 4.5 | Student zna zasady proceduralnego/obiekowego programowania i potrafi stworzyć złożony program w języku C/C++, obsługiwać We/Wy do/z pliku binarnego, opracować w programie osobne systemy obsługi błędów i komunikatów, oddzielić obsługę kontenera od obsługi danych. |
| NA OCENĘ 5.0 | Student zna zasady proceduralnego/obiekowego programowania i potrafi stworzyć złożony program w języku C/C++, obsługiwać We/Wy do/z pliku binarnego, opracować w programie osobne systemy obsługi błędów i komunikatów, oddzielić obsługę kontenera od obsługi danych, umie tworzyć szablony klas i funkcji, stosować wszystkie narzędzia polimorfizmu dynamicznego. |
| EFEKT KSZTAŁCENIA 2 | |
| NA OCENĘ 2.0 | Student nie spełnia warunków określonych dla oceny 3.0 |

| | |
|---------------------|---|
| NA OCENĘ 3.0 | Student nie zna wielu elementów składni, ale potrafi je zastąpić innymi; potrafi napisać prosty program w wersji proceduralnej/obiektywnej, ma kłopoty ze zrozumieniem działania programów w C/C++ na podstawie analizy kodu źródłowego napisanego przez innych programistów. |
| NA OCENĘ 3.5 | Student opanował składnię C/C++ w stopniu pozwalającym na zwarty zapis kodu prostych programów, potrafi analizować kod prostych programów, nie potrafi napisać programu w stylu obiektywnym. |
| NA OCENĘ 4.0 | Student potrafi napisać definicję struktury/klasy/unii i stworzyć tablicę obiektów złożonych. |
| NA OCENĘ 4.5 | Student zna wszystkie elementy składni C/C++, potrafi zaprojektować program używając styl obiektywny, umie łączyć dane w struktury/klasy/unii oraz panuje arytmetyką wskaźników. |
| NA OCENĘ 5.0 | Student zna składnię C/C++ w stopniu pozwalającym na tworzenie zwartych, szybkich i poprawnie działających programów, potrafi łączyć dane w struktury/klasy, unie, dynamicznie alokować i zwalniać pamięć dla tablic obiektów dowolnego typu, umie rozdzielić funkcjonalność kontenera od obsługi danych, umie stosować elementy STL. |
| EFEKT KSZTAŁCENIA 3 | |
| NA OCENĘ 2.0 | Student nie spełnia warunków określonych dla oceny 3.0 |
| NA OCENĘ 3.0 | Student rozumie działanie programu na podstawie lektury kodu źródeł. |
| NA OCENĘ 3.5 | Potrafi napisać program w C/C++ realizujący przyjęte założenia. |
| NA OCENĘ 4.0 | Potrafi zrobić analizę pozwalającą ustalić wymagania jakie stawiamy programowi. |
| NA OCENĘ 4.5 | Potrafi zastosować kod używający dynamicznych struktur danych. |
| NA OCENĘ 5.0 | Potrafi napisać kod optymalny pod względem wydajności i uzasadnić jego optymalność. |
| EFEKT KSZTAŁCENIA 4 | |
| NA OCENĘ 2.0 | Student nie spełnia warunków określonych dla oceny 3.0 |
| NA OCENĘ 3.0 | Potrafi napisać definicję prostej struktury/klasy bez odniesienia się do potrzeb programu. |
| NA OCENĘ 3.5 | Student umie zaprojektować strukturę, klasę, unie, enumeration dopasowane do potrzeb programu. |
| NA OCENĘ 4.0 | Potrafi tworzyć własne biblioteki funkcji. |
| NA OCENĘ 4.5 | Potrafi zrobić projekt współdziałania wszystkich struktur, klas i unii współdziałających w programie, potrafi zamykać dane i funkcje lokalne w pliku i udostępniać działania nad danymi poprzez tworzenie funkcji interfejsowych w języku C oraz stosować zasady enkapsulacji w C++. |

| | |
|--------------|---|
| NA OCENĘ 5.0 | Student potrafi zaprojektować struktury, klasy, unie oraz ich obsługę w stylu obiektowym (język C), które modelują rzeczywiste złożone obiekty, umie definiować struktury/klasy, elementami których występują obiekty innych struktur/klas. W C++ potrafi tworzyć hierarchie klas w oparciu na dziedziczenie, stosować szablony funkcji i klas oraz implementować polimorfizm dynamiczny. |
|--------------|---|

10 MACIERZ REALIZACJI PRZEDMIOTU

| EFEKT KSZTAŁCENIA | ODNIESIENIE DANEGO EFEKTU DO SZCZEGÓŁOWYCH EFEKTÓW ZDEFINIOWANYCH DLA PROGRAMU | CELE PRZEDMIOTU | TREŚCI PROGRAMOWE | NARZĘDZIA DYDAKTYCZNE | SPOSOBY OCENY |
|-------------------|--|----------------------------|---|-----------------------|---------------|
| EK1 | I1_W06 I1_W08 I1_W11 | Cel 1 Cel 2 Cel 3 Cel 4 | P1 P2 P3 P4 P5 P6 W1 W2 W3 W4 W5 W6 W7 W8 W9 W10 W11 W18 W19 W20 W21 K1 K2 K3 K4 K5 K6 K7 K8 K9 K13 K14 K15 K16 K19 K20 K24 K25 K26 | N1 N3 | F2 |
| EK2 | I1_W06 I1_W08 | Cel 2 Cel 3 Cel 4 | P1 P2 P3 P4 P5 P6 W1 W2 W3 W4 W5 W6 W7 W8 W9 W10 W11 W12 W13 W14 W15 W16 W17 W18 W19 W20 W21 W22 W23 W24 W25 W26 W27 W28 W29 K1 K2 K3 K4 K5 K6 K7 K8 K9 K10 K11 K12 K13 K14 K15 K16 K17 K18 K19 K20 K21 K22 K23 K24 K25 K26 K27 K28 | N1 N2 N3 | F1 F3 |

| EFEKT KSZTAŁCENIA | ODNIESIENIE DANEGO EFEKTU DO SZCZEGÓŁOWYCH EFEKTÓW ZDEFINIOWANYCH DLA PROGRAMU | CELE PRZEDMIOTU | TREŚCI PROGRAMOWE | NARZĘDZIA DYDAKTYCZNE | SPOSOBY OCENY |
|-------------------|--|-----------------|--|-----------------------|---------------|
| EK3 | I1_U07b I1_U10 | Cel 2 | W2 W3 W4 W5 W6 W7 W8 W9 W10 W11 W17 W18 W19 W20 W21 W22 W23 W24 K1 K2 K3 K4 K5 K6 K7 K8 K9 K10 K11 K12 K13 K14 K15 K16 K17 K18 K19 | N1 N2 N3 | F1 F3 |
| EK4 | I1_U01b I1_U10 I1_U22 | Cel 3 Cel 4 | P1 P2 P3 P5 P6 W1 W2 W3 W11 W15 W16 W17 W18 W19 W20 W21 W22 W23 W24 W25 W28 W29 K13 K14 K15 K16 K17 K18 K19 K20 K21 K22 K23 K24 K25 K26 K27 K28 | N1 N2 N3 | F1 |

11 WYKAZ LITERATURY

LITERATURA PODSTAWOWA

- [1] | B.Kernighan, D.Ritchie — *Język ANSI C*, Warszawa, 2002, WNT
- [2] | J. Richter, C. Nasarre. — *Windows via C/C++*, Microsoft Press, 2008, APN PROMISE
- [3] | B. Stroustrup, Piwko Łukasz — *Język C++. Kompendium wiedzy*, Warszawa, 2012, Helion
- [4] | H. Schildt — *Programowanie C++*, Warszawa, 2002, WNT

LITERATURA UZUPEŁNIAJĄCA

- [1] | S.Prata — *Język C. Szkoła Programowania*, Gliwice, 2006, Helion
- [2] | — *MSDN - Microsoft Developer Network*, , 2023, Dokumentacja Microsoft

12 INFORMACJE O NAUCZYCIELACH AKADEMICKICH

OSOBA ODPOWIEDZIALNA ZA KARTĘ

dr hab. inż. prof.PK. Sergiej Fialko (kontakt: sfialko@pk.edu.pl)

OSOBY PROWADZĄCE PRZEDMIOT

1 dr hab. inż. Siergiy Fialko (kontakt: sfialko@pk.edu.pl)

2 mgr. inż Jan Wojtas (kontakt: jwojtas@pk.edu.pl)

3 mgr. inż Katarzyna Smelcerz (kontakt: katarzyna.smelcerz@pk.edu.pl)

4 mgr. inż Krzysztof Swaldek (kontakt: krzysztof.swaldek1@pk.edu.pl)

5 mgr. inż Wojciech Książek (kontakt: wojciech.ksiazek@pk.edu.pl)

13 ZATWIERDZENIE KARTY PRZEDMIOTU DO REALIZACJI

(miejscowość, data)

(odpowiedzialny za przedmiot)

(dziekan)

PRZYJMUJĘ DO REALIZACJI (data i podpisy osób prowadzących przedmiot)

.....
.....
.....
.....
.....