**Nazwa przedmiotu:**

Programowanie obiektowe (I)

**Koordynator przedmiotu:**

Andrzej PAJĄK

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Automatyka i Robotyka

**Grupa przedmiotów:**

Przedmioty techniczne

**Kod przedmiotu:**

PROI

**Semestr nominalny:**

2 / rok ak. 2017/2018

**Liczba punktów ECTS:**

5

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

122

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład: | 30h |
| Ćwiczenia: | 0h |
| Laboratorium: | 30h |
| Projekt: | 0h |
| Lekcje komputerowe: | 0h |

**Wymagania wstępne:**

Brak

**Limit liczby studentów:**

100

**Cel przedmiotu:**

Praktyczne wprowadzenie do metodyki programowania obiektowego i rodzajowego z wykorzystaniem języka C++ i jego biblioteki standardowej.

**Treści kształcenia:**

Treść wykładu
Ewolucja metod programowania (2h).
Przykład rozwiązania problemu algorytmicznego z wykorzystaniem różnych stylów programowania: proceduralnego, obiektowego, generycznego. Zasada ukrywania szczegółów, abstrakcyjne typy danych. Podstawowe pojęcia paradygmatu programowania obiektowego (obiekty, klasy, dziedziczenie, polimorfizm, interfejs publiczny). Informacje wprowadzające o programowaniu rodzajowym. Język C++ na tle głównych metod programowania.
Mechanizmy pomocnicze języka C++ (4h).
System typów, typy wbudowane, typy referencyjne, mechanizmy tworzenia nowych typów; podstawy współpracy z WE/WY strumieniowym. Przeciążenia funkcji i operatorów (polimorfizm statyczny), prototypy funkcji, listy parametrów formalnych, parametry predefiniowane; funkcje rozwijane. Zarządzanie obiektami, tworzenie obiektów dynamicznych. Zakresy interpretacji nazw, przestrzenie nazw, operator '::'.
Klasy autonomiczne (4h).
Definiowanie klas samodzielnych, składowe klasy, kontrola dostępu, zaprzyjaźnienia. Konstruktory, destruktor i akcesoria syntetyzowane. Różnice pomiędzy klasami z wyróżnikiem class, struct i union. Projektowanie klas autonomicznych, przykłady. Składowe statyczne. Klasy ze zmienną strukturą wewnętrzną, postać kanoniczna. Klasy zagnieżdżone, wzorzec obiektu ze zliczaniem referencji. Listy inicjacyjne konstruktorów.
Szablony i podstawy programowania rodzajowego (4h).
Klasy parametryczne. Definiowanie szablonów klas i funkcji, parametryzacja szablonów. Konkretyzacje (instancje) szablonów. Specjalizacje szablonów: implikowane konkretyzacją, jawne częściowe i jawne pełne. Funkcje składowe i zaprzyjaźnione szablonowe. Podstawowe zasady programowania rodzajowego.
Dziedziczenie i polimorfizm (4h).
Dziedziczenie bezpośrednie, pośrednie, pojedyńcze, wielobazowe. Klasy bazowe wirtualne. Dziedziczenie a zawieranie. Polimorfizm dynamiczny, funkcje wirtualne, mechanizm aktywacji funkcji wirtualnych; obiekty polimorficzne. Funkcje wirtualne czyste, klasy abstrakcyjne i interfejsy. Szablony w dziedziczeniu. Model obiektu w C++. Tworzenie i destrukcja obiektów polimorficznych.
Polimorfizm i RTTI (2h).
Mechanizmy C++ do identyfikacji typów w czasie wykonania programu. Kontrolowane konwersje polimorficzne, zastosowania operatorów dynamic\_cast i typeid; wyjątek bad\_cast.
Obsługa sytuacji wyjątkowych (2h).
Mechanizm reagowania na sytuacje wyjątkowe, składnia i semantyka bloku try i bloków obsługi catch, aktywowanie wyjątku przez throw. Funkcje standardowe terminate(), unexpected (), set\_terminate (), set\_unexpected (). Zasady projektowania programów bezpiecznych, klasa szablonowa auto\_ptr.Wyjątki standardowe.
Biblioteka standardowa C++ (2h): Podstawowe akcesoria biblioteki - kontenery, iteratory i algorytmy. Klasyfikacja usług biblioteki standardowej; przegląd kontenerów sekwencyjnych i asocjacyjnych. Wsparcie dla współpracy ze strumieniami i przetwarzania tekstów. Podstawowe algorytmy generyczne.
Informacje uzupełniające i podsumowanie (2h).
Nowy standard C++11 (ISO/IEC 14882:2011), najważniejsze rozszerzenia w języku i bibliotece standardowej, perspektywy w najbliższych latach.
Zakres laboratorium
Studenci opracowują 3 projekty – programy o rosnącej skali trudności uwzględniające istotne aspekty programowania obiektowego. Problemy do rozwiązania obejmują: projektowanie klas autonomicznych, przeciążenie funkcji i operatorów, projektowanie klas ze zmienną strukturą obiektów, wykorzystanie reprezentacji grupowej obiektów ze zliczaniem referencji, definiowanie szablonów klas i funkcji, wykorzystanie dziedziczenia i funkcji wirtualnych, obsługę sytuacji wyjątkowych, współpracę ze strumieniami, projektowanie specjalizowanych klas kontenerowych i iteratorów a także wykorzystanie akcesoriów biblioteki standardowej.

**Metody oceny:**

Dwa sprawdziany audytoryjne; ocena 3 projektów w ramach laboratorium.

**Egzamin:**

nie

**Literatura:**

1. B. Stroustrup: Język C++, WNT, 2000, 2002.
2. S. B. Lippman: Podstawy języka C++, WNT 2001, 2003.
3. S. B. Lippman: Istota języka C++, WNT, 2004.
4. A. Koenig, B. Moo: C++ : Potęga języka, Helion, 2004.
5. B. Stroustrup: Programowanie. Teoria i praktyka z wykorzystaniem C++, Helion, 2010.

**Witryna www przedmiotu:**

https://studia.elka.pw.edu.pl/priv/12L/

**Uwagi:**

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt PROI\_W01:**

ma uporządkowaną wiedzę na temat podstawowych pojęć związanych z metodyką programowania obiektowego i generycznego

Weryfikacja:

sprawdziany; projekty 1..3

**Powiązane efekty kierunkowe:** K\_W03

**Powiązane efekty obszarowe:** T1A\_W02, T1A\_W03, T1A\_W07

**Efekt PROI\_W02:**

zna system typów i model obiektów języka C++

Weryfikacja:

sprawdziany; projekty 1..3

**Powiązane efekty kierunkowe:** K\_W03

**Powiązane efekty obszarowe:** T1A\_W02, T1A\_W03, T1A\_W07

**Efekt PROI\_W03:**

zna mechanizmy C++ wspierające metodykę obiektową (przeciążanie funkcji i operatorów, kontrola dostępu, funkcje wirtualne, RTTI, klasy abstrakcyjne, dziedziczenie wielobazowe)

Weryfikacja:

sprawdziany; projekty 1..3

**Powiązane efekty kierunkowe:** K\_W03

**Powiązane efekty obszarowe:** T1A\_W02, T1A\_W03, T1A\_W07

**Efekt PROI\_W04:**

zna mechanizmy pomocnicze (reguły zasięgu i przestrzenie nazw, wiązanie parametrów funkcji, funkcje rozwijane, obsługa wyjątków)

Weryfikacja:

sprawdziany; projekty 1..3

**Powiązane efekty kierunkowe:** K\_W03

**Powiązane efekty obszarowe:** T1A\_W02, T1A\_W03, T1A\_W07

**Efekt PROI\_W05:**

zna mechanizmy C++ wspierających programowanie generyczne (szablony funkcji i klas i ich cechy)

Weryfikacja:

sprawdziany; projekty 1..3

**Powiązane efekty kierunkowe:** K\_W03

**Powiązane efekty obszarowe:** T1A\_W02, T1A\_W03, T1A\_W07

**Efekt PROI\_W06:**

ma wiedzę na temat organizacji i zawartości biblioteki standardowej C++

Weryfikacja:

sprawdziany; projekty 1..3

**Powiązane efekty kierunkowe:** K\_W03

**Powiązane efekty obszarowe:** T1A\_W02, T1A\_W03, T1A\_W07

### Profil ogólnoakademicki - umiejętności

**Efekt PROI\_U01:**

potrafi zaprojektować klasę realizującą abstrakcyjny typ danych o określonych własnościach i interfejsie

Weryfikacja:

projekty 1..3

**Powiązane efekty kierunkowe:** K\_U01, K\_U02, K\_U07, K\_U10, K\_U21

**Powiązane efekty obszarowe:** T1A\_U01, T1A\_U02, T1A\_U09, T1A\_U07, T1A\_U14, T1A\_U15, T1A\_U16

**Efekt PROI\_U02:**

potrafi zmodyfikować istniejącą klasę (np. biblioteczną) przez zastosowanie dziedziczenia i podmiany funkcji wirtualnych

Weryfikacja:

projekty 1..3

**Powiązane efekty kierunkowe:** K\_U01, K\_U02, K\_U07, K\_U10, K\_U21

**Powiązane efekty obszarowe:** T1A\_U01, T1A\_U02, T1A\_U09, T1A\_U07, T1A\_U14, T1A\_U15, T1A\_U16

**Efekt PROI\_U03:**

potrafi zaproponować rozwiązanie nietrywialnego problemu algorytmicznego wykorzystując środki obiektowe języka C++ i jego biblioteki standardowej oraz zaplanować testy programu

Weryfikacja:

projekty 1..3

**Powiązane efekty kierunkowe:** K\_U01, K\_U02, K\_U07, K\_U10, K\_U21

**Powiązane efekty obszarowe:** T1A\_U01, T1A\_U02, T1A\_U09, T1A\_U07, T1A\_U14, T1A\_U15, T1A\_U16

**Efekt PROI\_U04:**

potrafi krytycznie ocenić rozwiązanie proceduralne problemu algorytmicznego i zaproponować lepsze rozwiązanie obiektowe

Weryfikacja:

projekty 1..3

**Powiązane efekty kierunkowe:** K\_U01, K\_U02, K\_U07, K\_U10, K\_U21

**Powiązane efekty obszarowe:** T1A\_U01, T1A\_U02, T1A\_U09, T1A\_U07, T1A\_U14, T1A\_U15, T1A\_U16

**Efekt PROI\_U05:**

potrafi przestrzegać przyjętego stylu dokumentowania i kodowania programów

Weryfikacja:

projekty 1..3

**Powiązane efekty kierunkowe:** K\_U01, K\_U02, K\_U07, K\_U10, K\_U21

**Powiązane efekty obszarowe:** T1A\_U01, T1A\_U02, T1A\_U09, T1A\_U07, T1A\_U14, T1A\_U15, T1A\_U16

**Efekt PROI\_U06:**

potrafi opracować klarowne sprawozdanie z wykonanego projektu programistycznego

Weryfikacja:

projekt 3

**Powiązane efekty kierunkowe:** K\_U01, K\_U02, K\_U07, K\_U10, K\_U21

**Powiązane efekty obszarowe:** T1A\_U01, T1A\_U02, T1A\_U09, T1A\_U07, T1A\_U14, T1A\_U15, T1A\_U16

### Profil ogólnoakademicki - kompetencje społeczne

**Efekt PROI\_K01:**

potrafi planować działania projektowe wg wymaganego terminu

Weryfikacja:

projekty

**Powiązane efekty kierunkowe:** K\_K04

**Powiązane efekty obszarowe:** T1A\_K04

**Efekt PROI\_K02:**

potrafi samodzielnie pozyskiwać uzupełniające informacje o środowisku języka C++

Weryfikacja:

projekty

**Powiązane efekty kierunkowe:** K\_K01

**Powiązane efekty obszarowe:** T1A\_K01

**Efekt PROI\_K03:**

rozumie rolę standaryzacji języka programowania

Weryfikacja:

projekty

**Powiązane efekty kierunkowe:** K\_K02

**Powiązane efekty obszarowe:** T1A\_K02