**Nazwa przedmiotu:**

Języki i techniki programowania cz. 2

**Koordynator przedmiotu:**

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Geoinformatyka

**Grupa przedmiotów:**

Obowiązkowe

**Kod przedmiotu:**

1060-GI000-ISP-2008

**Semestr nominalny:**

2 / rok ak. 2019/2020

**Liczba punktów ECTS:**

4

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

1) Liczba godzin kontaktowych - 47 godz., w tym: a) uczestnictwo w wykładach (15 godz.), b) uczestnictwo w ćwiczeniach (30 godz.), c) udział w konsultacjach (2 godz.). 2) Praca własna studenta - 60 godz. w tym: a) przygotowanie do ćwiczeń (15 godz.), b) samodzielna praca projektowa (30 godz.), c) przygotowanie do egzaminu (15 godz.). RAZEM: 107 godz. co odpowiada 4 pkt. ECTS.

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

1,8 pkt. ECTS - liczba godzin kontaktowych - 47 godz., w tym: a) uczestnictwo w wykładach (15 godz.), b) uczestnictwo w ćwiczeniach (30 godz.), c) udział w konsultacjach (2 godz.).

**Język prowadzenia zajęć:**

polski

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

2,2 pkt. ECTS - 60 godz., w tym: a) uczestnictwo w ćwiczeniach (30 godz.), b) samodzielna praca projektowa (30 godz.).

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

|  |  |
| --- | --- |
| Wykład: | 15h |
| Ćwiczenia: | 30h |
| Laboratorium: | 0h |
| Projekt: | 0h |
| Lekcje komputerowe: | 0h |

**Wymagania wstępne:**

Znajomość podstaw informatyki i techniki komputerowej, umiejętność programowania strukturalnego, znajomość podstawowych algorytmów i struktur danych wykorzystywanych w programowaniu

**Limit liczby studentów:**

-

**Cel przedmiotu:**

Celem przedmiotu jest zapoznanie studentów z programowaniem obiektowym, wzorcami projektowymi oraz metodami tworzenia interfejsów graficznych

**Treści kształcenia:**

Wykłady:
1. Dziedziczenie: klasy bazowe i pochodne. Możliwości identyfikacji klasy obiektu w czasie kompilacji i wykonania programu. Drzewo dziedziczenia i konwersje typów.
2. Dziedziczenie: metody wirtualne, metody czysto wirtualne i klasy abstrakcyjne. Znaczenie destruktorów wirtualnych.
3. Dziedziczenie, agregacja, kompozycja. Zasady stosowania poszczególnych rozwiązań.
4. Wykorzystanie dziedziczenia, obiektów abstrakcyjnych w programowaniu elementów graficznych.
5. Wzorce projektowe: podstawy. Wzorce czynnościowe: iterator, obserwator, odwiedzający, metoda szablonowa.
6. Strukturalne wzorce projektowe: fasada, dekorator, adapter.
7. Konstrukcyjne wzorce projektowe: singleton, metoda wytwórcza, fabryka abstrakcyjna.
8. Zasady wykorzystania wzorców projektowych; adaptacja do konkretnych potrzeb.
9. Podstawy graficznych interfejsów użytkownika: zdarzenia i ich obsługa.
10. Typowe elementy graficznego interfejsu użytkownika i ich parametry.
11. Separacja warstwy prezentacji oraz obsługi zdarzeń. Polecenia.
12. Podstawy wzorca model-widok-kontroler. Przykłady realizacji wzorca MVC.
13. Wykorzystanie wzorca MVC do tworzenia aplikacji.
14. Automatyzacja procesu tworzenia oprogramowania: generatory kodu. Narzędzia wspomagające pracę programisty: system kontroli wersji, system ciągłej integracji.
15. Podstawowe informacje dotyczące innych, powszechnie używanych języków programowania i środowisk programowania.
Ćwiczenia:
1. Prosta rodzina klas z wykorzystaniem dziedziczenia.
2. Wykorzystanie czynnościowych wzorców projektowych w prostym programie z graficzną prezentacją wyników działania.
3. Program wykorzystujący konstrukcyjne wzorce projektowe i hierarchię klas.
4. Prosty program z graficznym interfejsem użytkownika.
5. Program wykorzystujący strukturę model-widok-kontroler do obsługi dokumentów o różnej strukturze.

**Metody oceny:**

W trakcie zajęć student może zgromadzić maksymalnie 100 punktów podzielonych na: 50 punktów z zadań programistycznych ocenianych w skali 0-10, 40 punktów za egzamin przeprowadzony w formie komputerowej oraz 10 punktów za realizację dodatkowych zadań domowych. Do zaliczenia przedmiotu trzeba uzyskać minimum 50% punktów z egzaminu (ocena z wykładu) oraz zadań programistycznych (ocena z ćwiczeń). Punkty są przeliczane na oceny wg następujących zakresów: 0-50 punktów – ocena 2, 51-60 punktów – ocena 3.0, 61-70 punktów – ocena 3.5, 71-80 punktów - ocena 4.0, 81-90 punktów – ocena 4.5, 91-100 punktów – ocena 5.0.

**Egzamin:**

tak

**Literatura:**

1) Bjarne Stroustrup, Programowanie. Teoria i praktyka z wykorzystaniem C++, Helion, 2013.
2) Robert Nowak, Andrzej Pająk, Język C++ mechanizmy, wzorce, biblioteki, btc, 2013.
3) Erich Gamma, Richard Helm, Ralph Johnson, John M. Vlissides, Wzorce projektowe. Elementy oprogramowania obiektowego wielokrotnego użytku, Helion, 2010

**Witryna www przedmiotu:**

-

**Uwagi:**

Oprogramowanie wykorzystywane na ćwiczeniach: zestaw kompilatorów i narzędzi GCC (MinGW), zintegrowane środowisko Eclipse CDT

## Charakterystyki przedmiotowe

### Profil praktyczny - wiedza

**Charakterystyka 1060-GI000-ISP-2008\_W01:**

Zna zasady i mechanizmy programowania obiektowego

Weryfikacja:

Egzamin

**Powiązane charakterystyki kierunkowe:** K\_W04

**Powiązane charakterystyki obszarowe:** I.P6S\_WG

**Charakterystyka 1060-GI000-ISP-2008\_W02:**

Zna podstawowe wzorce projektowe i zasady ich wykorzystania

Weryfikacja:

Egzamin

**Powiązane charakterystyki kierunkowe:** K\_W04

**Powiązane charakterystyki obszarowe:** I.P6S\_WG

**Charakterystyka 1060-GI000-ISP-2008\_W03:**

Zna metody tworzenia graficznego interfejsu użytkownika

Weryfikacja:

Egzamin

**Powiązane charakterystyki kierunkowe:** K\_W04

**Powiązane charakterystyki obszarowe:** I.P6S\_WG

**Charakterystyka 1060-GI000-ISP-2008\_W04:**

Ma podstawową wiedzę w zakresie automatyzacji procesu tworzenia i testowania oprogramowania

Weryfikacja:

Egzamin

**Powiązane charakterystyki kierunkowe:** K\_W03, K\_W04

**Powiązane charakterystyki obszarowe:** I.P6S\_WG

### Profil praktyczny - umiejętności

**Charakterystyka 1060-GI000-ISP-2008\_U01:**

Potrafi zaprojektować i napisać program w sposób obiektowy

Weryfikacja:

Zadanie programistyczne

**Powiązane charakterystyki kierunkowe:** K\_U01, K\_U06, K\_U15

**Powiązane charakterystyki obszarowe:** I.P6S\_UW, I.P6S\_UU

**Charakterystyka 1060-GI000-ISP-2008\_U02:**

Potrafi zastosować wzorce projektowe we własnym programie

Weryfikacja:

Zadanie programistyczne

**Powiązane charakterystyki kierunkowe:** K\_U01, K\_U06, K\_U15

**Powiązane charakterystyki obszarowe:** I.P6S\_UW, I.P6S\_UU

**Charakterystyka 1060-GI000-ISP-2008\_U03:**

Potrafi skorzystać z bibliotek i ich dokumentacji

Weryfikacja:

Zadanie programistyczne

**Powiązane charakterystyki kierunkowe:** K\_U15, K\_U01, K\_U06, K\_U10

**Powiązane charakterystyki obszarowe:** I.P6S\_UW, I.P6S\_UU

**Charakterystyka 1060-GI000-ISP-2008\_U04:**

Potrafi zaprojektować i napisać program z prostym graficznym interfejsem użytkownika

Weryfikacja:

Zadanie programistyczne

**Powiązane charakterystyki kierunkowe:** K\_U01, K\_U06, K\_U10, K\_U15

**Powiązane charakterystyki obszarowe:** I.P6S\_UW, I.P6S\_UU

### Profil praktyczny - kompetencje społeczne

**Charakterystyka 1060-GI000-ISP-2008\_K01:**

Potrafi współpracować w zespole rozwiązując przydzielone problemy

Weryfikacja:

Zadanie programistyczne

**Powiązane charakterystyki kierunkowe:** K\_K01, K\_K04

**Powiązane charakterystyki obszarowe:** I.P6S\_KK, I.P6S\_KO, I.P6S\_KR