

Opracował: Włodzimierz Kasprzak

Standardy przedmiotu w ramach programu nauczania

1. Nazwa przedmiotu: **Języki i metody programowania (sem. 3)**

Podstawy programowania obiektowego.

2. Semestr nauczania: trzeci

3. Forma zaliczania: zaliczenie

4. Liczba godzin:

- wykładów: 14 (D), 10 (W), 10 (Z)
- ćwiczeń (laboratoriów): 28 (D), 20 (W), 20 (Z)

5. Konspekt: Tworzenie aplikacji programowej. Paradygmat programowania obiektowego. Klasy, obiekty, interfejsy. Funkcje i przekazywanie argumentów. Przeciążanie funkcji. Zasięgi widoczności nazw. Dziedziczenie klas. Przesłanianie metod. Metody wirtualne. Obsługa wyjątków. System wejścia – wyjścia. Program w środowisku SDK (Delphi 6/7 – „Object Pascal” lub MS .NET – „C#” lub Sun ONE Studio – „Java”). Pojęcie własności i publikowania własności. Kontrolki – klasy obiektów „wizualnych”. Zdarzenia i mechanizm ich obsługi.

6. Zakres minimalnych wymagań dotyczących wiedzy i umiejętności po ukończeniu przedmiotu przez studenta:

- Zasady programowania obiektowego – definicja klasy, rodzaje klas, obiekt i czas życia obiektu, dziedziczenie, polimorfizm, przeciążanie funkcji, przesłanianie metod, implementacja interfejsu.
- Mechanizm obsługi wyjątków.
- Obiektowe wejście-wyjście.
- Przekazywanie argumentów do funkcji.
- Tworzenie obiektowych aplikacji konsolowych w środowisku SDK.

7. Plan realizacji wykładu:

Plan realizacji programu nauczania (na przykładzie Delphi)		
Nr tyg.	Wykłady	Pojęcia wprowadzane podczas wykładu
1.	Wstęp. Paradygmat programowania obiektowego.	Tworzenie programu w Delphi – aplikacja konsolowa i okienkowa. Abstrakcyjny typ danych i klasa. Obiekt. Dziedziczenie klas. Polimorfizm – wielopostaciowość. Enkapsulacja – kapsułkowanie. Przegląd głównych języków programowania obiektowego.

2.	Obiekty. Dziedziczenie klas.	Tworzenie i usuwanie obiektów – cykl życia obiektu. Kwalifikacje dostępu do składowych. Metody klasowe. Dziedziczenie. Przesłanianie a odslanianie składowych dziedziczonych. Konwersje typów obiektu.
3.	Zasięgi interpretacji nazw. Funkcje – argumenty, przeciążanie.	Struktura modułu programu. Zasięgi interpretacji nazw. Sposoby przekazywania argumentów do funkcji. Typ funkcji. Przeciążanie nazwy funkcji.
4.	Hierarchia klas w SDK (np. Delphi) Przykłady aplikacji.	Główne klasy w Delphi 7. Klasy TPersistent i TControl. Klasy głównych kontrolerek.
5.	<i>Sprawdzian (pkt 1-8).</i> Właściwości klasy.	Pojęcie <i>property</i> i jego mechanizm.
6.	Dynamiczna konwersja typów. Zdarzenia.	Operatory <i>is, as</i> . Reprezentacja zdarzeń, typy procedur obsługi, podstawowe klasy zdarzeń w Delphi.
7.	Obsługa wyjątków. Obiektowe wejście-wyjście.	<i>try, catch, finally, raise</i> , klasa <i>Exception</i> , Obiektowa współpraca we-wy z plikami, tablicami wewnętrznymi i komponentami Delphi - klasa <i>TStream</i> i jej klasy pochodne.

8. Wymagania dotyczące laboratoriów: W jednym ze środowisk SDK (Delphi 6/7 – „Object Pascal” lub MS .NET – „C#” lub Sun ONE Studio – „Java”) - tworzenie aplikacji konsolowych i okienkowych wykorzystujących podstawowe pojęcia programowania obiektowego: klasa, obiekt, pola i funkcje składowe, konstrukcja i destrukcja obiektu, dziedziczenie klas, przesłanianie i odslanianie składowych, metody wirtualne, dynamiczne konwersje typów, sposoby przekazywania argumentów w wywołaniu funkcji, własności i publikowanie własności, reprezentacja zdarzeń, obsługa sytuacji wyjątkowych, obiektowe wejście – wyjście, wykorzystanie obiektów okienkowych i kontrolerek.

9. Literatura podstawowa (na przykładzie Delphi):

1. W. Kasprzak: Materiały elektroniczne do wykładu „Języki i metody programowania – podstawy programowania obiektowego w „Object Pascal””, WS Miła College, Warszawa, 2002-2003.
2. Marco Cantu: *Delphi 6: praktyka programowania*. Wyd. MIKOM, Warszawa, 2002.

10. Wymagane zagadnienia (umiejętności) niezbędne w realizacji danego przedmiotu: