



**113-17-5**

## **PROGRAMOWANIE**

**ECTS: 12**

## **PROGRAMMING**

### **TREŚCI WYKŁADÓW**

Przedstawienie istoty paradygmatu obiektowego: odpowiedzialność obiektu, ukrywanie informacji, abstrakcja systemu obiektowego, dziedziczenie, hermetyzacja, polimorfizm metod i polimorfizm parametryczny. Omówione są najważniejsze konstrukcje obiektowe języka C# realizujące istotę paradygmatu obiektowego: pojęcia klasy i obiektu, poziomy dostępu, dziedziczenie, metody wirtualne (polimorfizm metod), typy generyczne (polimorfizm parametryczny), klasy abstrakcyjne i interfejsy. Zastosowanie wprowadzonych konstrukcji programistycznych do hermetyzacji pól, metod, typów, obszarów zmienności kodu. Omówione są podstawy projektowania obiektowego. Wprowadzony jest podstawowy formalizm projektów programów obiektowych: diagramy klas. Zaprezentowanie kilku wybranych wzorców projektowych, z podkreśleniem roli wzorców projektowych jako sposobów hermetyzacji obszarów zmienności programu.

### **TREŚCI ĆWICZEŃ**

W ramach laboratorium studenci mają pisać i uruchamiać szereg małych programów ilustrujących kolejne zagadnienia przedstawiane na wykładzie.

### **CEL KSZTAŁCENIA**

1. Przedstawienie paradygmatu programowania obiektowego jako metodologii tworzenia programów 2. Tworzenie programów odpornych na błędy użytkownika 3. Poznanie środowiska programistycznego Visual Studio 2010

### **OPIS EFEKTÓW KSZTAŁCENIA PRZEDMIOTU W ODNIESIENIU DO OBSZAROWYCH I KIERUNKOWYCH EFEKTÓW KSZTAŁCENIA**

**Symbole efektów obszarowych** T1A\_W01, T1A\_W04, T1A\_W05, T1A\_W07, T1A\_U01, T1A\_U02, T1A\_U05, T1A\_K01, T1A\_K03

**Symbole efektów kierunkowych** K\_W01, K\_W07, K\_U01, K\_U02, K\_K01, K\_K02, K\_K04

### **EFEKTY KSZTAŁCENIA**

#### **Wiedza**

Student zna następujące obiektowe konstrukcje programistyczne: 1. Klasy i obiekty. Odpowiedzialności (kontrakty) klas. Pola i metody statyczne. Poziomy dostępu do pól i metod. 2. Dziedziczenie klas i metody polimorficzne (wirtualne). 3. Typy wartościowe i typy referencyjne. 4. Interfejsy i klasy abstrakcyjne. 5. Typy generyczne. Kolekcje (wbudowane i programowane). 6. Wyjątki w obiektowym języku programowania.

#### **Umiejętności**

Umiejętności kognitywne: 1. Student potrafi wyabstrahować ze sformułowania programowanego zadania system klas i obiektów (analiza modelowa). 2. Student potrafi wykryć relewantne poziomy abstrakcji w sformułowaniu programowanego zadania. 3. Student posiada umiejętność analizowania możliwych zmian wymagań dotyczących tworzonego programu. 4. Student potrafi uwzględnić wykryte możliwe zmiany wymagań, jako obszary zmienności w projektowanym programie obiektowym. Umiejętności praktyczne: 1. Student

#### **Kompetencje społeczne**

Student docenia rolę precyzji w formułowaniu problemów. Jest świadomy ważności poprawności tworzonego oprogramowania, zwłaszcza, gdy poprawność oprogramowania jest krytycznym warunkiem jego stosowania. Student jest także świadomy konieczności ciągłego doskonalenia swoich umiejętności programistycznych. Student wie, że właściwa komunikacja z członkami zespołu jest warunkiem koniecznym stworzenia dużego oprogramowania rozwiązującego stawiany problem. Student jest świadomy roli przestrzegania warun

### **LITERATURA PODSTAWOWA**

1) Adam Boduch, 2010r., "Wstęp do programowania w języku C#", wyd. Helion, t.1, s.392, 2) Klaus Michelsen, 2007r., "Język C#. Szkoła programowania", wyd. Helion, t.1, s.1128, 3) Marcin Lis, 2007r., "C#. Praktyczny kurs", wyd. Helion, t.1, s.376, 4) Mirosław Kubiak, 2012r., "C#. Zadania z programowania z przykładowymi rozwiązaniami", wyd. Helion, t.1, s.128.

**Przedmiot/moduł:**  
PROGRAMOWANIE

**Obszar kształcenia:** nauki techniczne

**Status przedmiotu:** Obligatoryjny

**Kod ECTS:** 113-17-5

**Nazwa studiów podyplomowych/kursu:**

Zaawansowane technologie informatyczne

**Forma studiów:** Niestacjonarne

**Poziom studiów/Forma kształcenia:** Studia

podyplomowe/kurs dokształcający

**Rok/semestr:** I/2

**Rodzaje zajęć:** wykład, ćwiczenia laboratoryjne

**Liczba godzin w semestrze**

wykłady: 20/2

ćwiczenia: 30/3

**Formy i metody dydaktyczne**

wykłady: Definiowanie i objaśnianie wprowadzanych konstrukcji programistycznych i zasad paradygmatu obiektowego.

**ćwiczenia:** projektowanie i programowanie

**Forma i warunki zaliczenia:** Zaliczenie na ocenę/

Wykład – obecność i aktywność, Ćwiczenia – dwa

kolokwia sprawdzające praktyczną umiejętność

implementacji poznanych metod.

**Liczba punktów ECTS:** 12

**Język wykładowy:** polski

**Wymagania wstępne:**

**Nazwa jednostki organizacyjnej realizującej**

**przedmiot:**

Katedra Informatyki i Badań Operacyjnych

**adres:** ul. Słoneczna 54, , 10-710 Olsztyn

tel. 524 60 92

**Osoba odpowiedzialna za realizację przedmiotu:**

mgr inż. Magdalena Modrzyńska

**e-mail:** magda@matman.uwm.edu.pl

## Szczegółowy opis przyznanej punktacji ECTS - część B

### PROGRAMOWANIE

**ECTS: 12**

### PROGRAMMING

Na przyznaną liczbę punktów ECTS składają się :

1. Godziny kontaktowe z nauczycielem akademickim:

- Godziny kontaktowe z nauczycielem akademickim

50,0 godz.

---

50,0 godz.

2. Samodzielna praca studenta:

- Samodzielna praca studenta

10,0 godz.

---

10,0 godz.

godziny kontaktowe + samodzielna praca studenta OGÓŁEM:

60,0 godz.

1 punkt ECTS = 5,00 godz. pracy przeciętnego studenta,

liczba punktów ECTS = 60,00 godz.: 5,00 godz./ECTS = **12,00 ECTS**

w zaokrągleniu: **12 ECTS**

- w tym liczba punktów ECTS za godziny kontaktowe z bezpośrednim udziałem nauczyciela akademickiego - **10,00** punktów ECTS,

- w tym liczba punktów ECTS za godziny realizowane w formie samodzielnej pracy studenta - **2,00** punktów ECTS.