KARTA OPISU MODUŁU KSZTAŁCENIA

Nazwa modułu/przedmiotu	Kod
Modelowanie i programowanie obiektowe	**1010515311010514676**

Kierunek studiów	Profil kształcenia (ogólnoakademicki, praktyczny)	Rok / Semestr
Informatyka	**ogólnoakademicki**	**1 / 1**
Ścieżka obieralności/specjalność	Przedmiot oferowany w języku:	Kurs (obligatoryjny/obieralny)
Sieci komputerowe	**polski**	**obligatoryjny**

Stopień studiów:	**Forma studiów** (stacjonarna/niestacjonarna)
II stopień	**niestacjonarna**

Godziny	Liczba punktów
Wykłady: **8** Ćwiczenia: **-** Laboratoria: **24** Projekty/seminaria: **-**	**4**

Status przedmiotu w programie studiów (podstawowy, kierunkowy, inny)	(ogólnouczelniany, z innego kierunku)
kierunkowy	**z danego kierunku**

Obszar(y) kształcenia i dziedzina(y) nauki i sztuki	**Podział ECTS (liczba i %)**
nauki techniczne	**4 100%**

Odpowiedzialny za przedmiot / wykładowca:

dr inż. Cezary Sobaniec
email: Cezary.Sobaniec@cs.put.poznan.pl
tel. 61 6652370
Instytut Informatyki
ul. Piotrowo 2, 60-965 Poznań

Wymagania wstępne w zakresie wiedzy, umiejętności, kompetencji społecznych:

1	**Wiedza:**	Student rozpoczynający ten przedmiot powinien posiadać podstawową wiedzę z zakresu algorytmów i struktur danych.
2	**Umiejętności:**	Ponadto, student powinien posiadać umiejętność rozwiązywania podstawowych problemów oraz samodzielnego pisania, modyfikowania i testowania programów komputerowych. Niezbędna jest również umiejętność pozyskiwania informacji ze wskazanych źródeł oraz rozumienie konieczności pracy w grupie nad wspólnym projektem.
3	**Kompetencje społeczne**	Dodatkowo student musi prezentować takie postawy społeczne jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista i szacunek dla innych ludzi.

Cel przedmiotu:

1. Nauczenie studentów zasad modelowania oraz tworzenia programów zgodnie z paradygmatem obiektowym, z uwzględnieniem podziału takich aplikacji na re-używalne komponenty oraz tworzeniem nowych typów danych.
2. Rozwijanie u studentów umiejętności modelowania i tworzenia systemów informatycznych o odpowiedniej architekturze, która cechuje się spoistością składowych modułów programowych i luźnych związków między tymi modułami.

Efekty kształcenia i odniesienie do kierunkowych efektów kształcenia

Wiedza:

1. ma uporządkowaną, podbudowaną teoretycznie wiedzę ogólną w zakresie algorytmów, architektury systemów komputerowych, języków i paradygmatów programowania oraz inżynierii oprogramowania, - [K_W4]
2. ma podbudowaną teoretycznie szczegółową wiedzę związaną z wybranymi zagadnieniami z zakresu informatyki - [K_W5]
3. ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach w informatyce i w wybranych pokrewnych dyscyplinach naukowych - [K_W6]
4. ma podstawową wiedzę o cyklu życia systemów informatycznych sprzętowych lub programowych - [K_W7]
5. zna podstawowe metody, techniki i narzędzia stosowane przy rozwiązywaniu złożonych zadań inżynierskich z wybranego obszaru informatyki - [K_W8]

Umiejętności:

1. potrafi pozyskiwać informacje z literatury, baz danych oraz innych źródeł (w języku ojczystym i angielskim), integrować je, dokonywać ich interpretacji i krytycznej oceny, wyciągać wnioski oraz formułować i wyczerpująco uzasadniać opinii - [K_U1]

2. potrafi - przy formułowaniu i rozwiązywaniu zadań inżynierskich - integrować wiedzę z różnych obszarów informatyki (a w razie potrzeby także wiedzę z innych dyscyplin naukowych) oraz zastosować podejście systemowe, uwzględniające także aspekty pozatechniczne - [K_U10]

3. potrafi ocenić przydatność i możliwość wykorzystania nowych osiągnięć (metod i narzędzi) oraz nowych produktów informatycznych - [K_U13]

4. potrafi stworzyć model obiektowy prostego systemu (np. w języku UML) - [K_U17]

5. potrafi wybrać język programowania odpowiedni do danego zadania programistycznego - [K_U26]

Kompetencje społeczne:

1. rozumie, że w informatyce wiedza i umiejętności bardzo szybko stają się przestarzałe - [K_K1]

2. zna przykłady i rozumie przyczyny wadliwie działających systemów informatycznych, które doprowadziły do poważnych strat finansowych, społecznych lub też do poważnej utraty zdrowia, a nawet życia - [K_K4]

3. potrafi odpowiednio określić priorytety służące realizacji określonego przez siebie lub innych zadania - [K_K6]

Sposoby sprawdzenia efektów kształcenia

Efekty kształcenia przedstawione wyżej weryfikowane są w następujący sposób:

Ocena formująca:

a) w zakresie wykładów:

- na podstawie odpowiedzi na pytania dotyczące materiału omówionego na poprzednich wykładach,

b) w zakresie zajęć laboratoryjnych:

- na podstawie oceny bieżącego postępu realizacji zadań,

Ocena podsumowująca:

- ocenę wiedzy i umiejętności wykazanych na egzaminie pisemnym o charakterze problemowym, składającym się z 3 zadań. Łączna liczba punktów, jaką można uzyskać za prawidłowe rozwiązanie zadań wynosi 3 punkty. Aby zaliczyć egzamin i uzyskać ocenę 3.0, student musi uzyskać co najmniej 50% maksymalnej liczby punktów (tj. 1,5 punktu). W trakcie egzaminu student nie może korzystać z materiałów dydaktycznych.
- omówienie wyników egzaminu,

b) w zakresie zajęć laboratoryjnych weryfikowanie założonych efektów kształcenia realizowane jest przez:

- ocenę przygotowania studenta do poszczególnych zajęć (sprawdzian ?wejściowy"),

Uzyskiwanie punktów dodatkowych za aktywność podczas zajęć, a szczególnie za:

- omówienia dodatkowych aspektów zagadnienia,
- efektywność zastosowania zdobytej wiedzy podczas rozwiązywania zadanego problemu,
- umiejętność współpracy w ramach zespołu praktycznie realizującego zadanie szczegółowe w laboratorium,
- uwagi związane z udoskonaleniem materiałów dydaktycznych,
- wskazywanie trudności percepcyjnych studentów umożliwiające bieżące doskonalenia procesu dydaktycznego.

Treści programowe

Program wykładów z przedmiotu obejmuje następujące zagadnienia.

Motywacje dla nowego paradygmatu oprogramowania ? programowanie obiektowe. Idea nowego paradygmatu programowania. Różnice pomiędzy podejściem obiektowym, a proceduralnym. Historia języków obiektowych. Podstawowe pojęcia związane z paradygmatem obiektowym. Założenia paradygmatu obiektowego.

Problem konstrukcji złożonych systemów informatycznych. Analiza i projektowanie obiektowe. Obiektowe modelowanie dziedziny. Modelowanie z wykorzystaniem języka UML. Tworzenie diagramów przebiegu. Mapowanie języka UML na C# i odwrotnie. Karty CRC.

Podstawowe konstrukcje paradygmatu obiektowego oraz idee. Budowa klasy. Budowa programu. Zakresy widoczności. Tworzenie i niszczenie obiektów. Własności programów obiektowych i ich konsekwencje. Mechanizm dziedziczenia. Polimorfizm. Składowe klasy.

Wprowadzenie do platformy .Net. Historia platformy i jej rozwoju. Alternatywne platformy na przykładzie Mono. Przedstawienie podstawowych konstrukcji obiektowych na przykładzie języka C#.

W ramach zajęć laboratoryjnych studenci zapoznają się z obiektowym językiem programowania: C# oraz podstawowymi funkcjami środowiska programistycznego Visual Studio. Ćwiczenia polegają na samodzielnym tworzeniu programów zawierających podstawowe konstrukcje języków obiektowych przedstawianych na wykładach i laboratorium. Dodatkowo, realizowane są niewielkie zadania domowe poszerzające wiedzę i umiejętności zdobyte na zajęciach.

Cześć wymienionych wyżej treści programowych realizowana jest w ramach pracy własnej studenta.

Metody dydaktyczne:

1. wykład: pokaz multimedialny, prezentacja ilustrowana przykładami podawanymi na tablicy,
2. ćwiczenia laboratoryjne: ćwiczenia praktyczne, dyskusja, praca w zespole, pokaz multimedialny

Literatura podstawowa:

1. Programowanie zorientowane obiektowo, Bertrand Mayer, Helion, Warszawa, 2005
2. C#. Leksykon, Ben Albahari, Peter Drayton, Brad Merrill, Helion , Gliwice, 2001
3. Wstęp do programowania w języku C#, Adam Boduch, Helion, Gliwice, 2006
4. http://msdn.microsoft.com/en-us/library/vstudio/kx37x362.aspx

Literatura uzupełniająca:

1. Metody obiektowe w teorii i praktyce, Ian Graham, WNT, Warszawa, 2004
2. Microsoft(R) Visual C#(R) 2010 Step by Step, John Sharp, Redmond, Washington , 2010

Bilans nakładu pracy przeciętnego studenta

Czynność	Czas (godz.)
1. udział w zajęciach laboratoryjnych:	24
2. przygotowanie do ćwiczeń laboratoryjnych:	24
3. realizacja zadań domowych:	12
4. udział w konsultacjach (mogą być realizowane drogą elektroniczną) związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych / zadań domowych	2
5. napisanie programu / programów, uruchomienie I weryfikacja (czas poza zajęciami laboratoryjnymi)	10
6. przygotowanie do egzaminu 8 godz. + egzamin 2godz.	10
7. udział w wykładach	8
8. zapoznanie się ze wskazaną literaturą / materiałami dydaktycznymi (10 stron tekstu naukowego = 1 godz.), 100 stron	10

Obciążenie pracą studenta

forma aktywności	godzin	ECTS
Łączny nakład pracy	100	4
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	36	1
Zajęcia o charakterze praktycznym	72	3