

Programowanie w środowiskach zintegrowanych

Kod przedmiotu: PSZ

Rodzaj przedmiotu: specjalnościowy; obieralny.

Wydział: Informatyki

Kierunek: Informatyka

Specjalność (specjalizacja): Inżynieria Systemów Informatycznych

Poziom studiów: pierwszego stopnia

Profil studiów: ogólnoakademicki

Forma studiów: stacjonarne, niestacjonarne

Rok: 3 **Semestr:** 5

Formy zajęć i liczba godzin:

w formie stacjonarnej: **wykłady – 30 ; ćwiczenia laboratoryjne – 30**

w formie niestacjonarnej: **wykłady – 20 ; ćwiczenia laboratoryjne – 20**

Język/i, w którym/ch realizowane są zajęcia: zajęcia w języku polskim

Liczba punktów ECTS: 5

Założenia i cele przedmiotu:

Przedmiot ten poświęcony jest rozwijaniu umiejętności tworzenia oprogramowania z wykorzystaniem najpopularniejszych środowisk programistycznych typu RAD (np. C++ Builder, Visual Studio, QtCreator). Kształcenie ukierunkowane jest na poznanie i praktyczne przećwiczenie budowania programów sterowanych zdarzeniami, wykorzystujących graficzny interfejs użytkownika oraz biblioteki zadaniowo zorientowanych klas i komponentów. Zakłada się, że po zaliczeniu tego przedmiotu student będzie potrafił zrealizować pełnowartościowe, nowoczesne oprogramowanie klasy desktop oraz oprogramowanie klienckie dla systemów w architekturze klient-serwer.

Określenie przedmiotów wprowadzających wraz z wymaganiami wstępnymi:

Programowanie w środowiskach zintegrowanych - to przedmiot na specjalizacji Inżyniera Systemów Informatycznych. Stanowi on kontynuację przedmiotów Podstawy Programowania, Języki programowania, Języki Programowania Obiektowego, Wymogi wstępne dotyczą wiedzy uzyskanej przez studentów właśnie w ramach przedmiotów Podstawy Programowania, Języki programowania i Języki Programowania Obiektowego.

Opis form zajęć

a) Wykłady

• Treści programowe (tematyka zajęć):

1. Programowanie sterowane przepływem a programowanie sterowane zdarzeniami.
2. Charakterystyka ogólna pakietów RAD
3. Budowa aplikacji w środowiskach RAD.

4. Projektowanie wizualne graficznego interfejsu użytkownika.
5. Programowanie sterowane zdarzeniami w systemach RAD.
6. Charakterystyka wybranych pakietów RAD:
7. Koncepcja organizacji kodu.
8. Elementy typowe dla pakietów RAD:
 - Modelowanie interfejsu użytkownika
 - Biblioteki komponentów graficznych i systemowych.
 - Biblioteki klas kontenerowych, dostępu do baz danych, odwołania do usług systemu operacyjnego.

• Metody dydaktyczne:

Wykład prowadzony metodą tradycyjną z wykorzystaniem rzutnika multimedialnego, obejmować będą również prezentację przykładów z wykorzystaniem wybranych środowisk programistycznych oraz ich dyskusje z aktywnym uczestnictwem studentów. Materiały wspomagające, uzupełniające i związane z pracą własną studenta udostępniane są w wersji elektronicznej.

• Forma i warunki zaliczenia:

Warunkiem zaliczenia całości przedmiotu jest zdanie egzaminu. Forma realizacji egzaminu dostosowywana jest do liczebności grupy studenckiej oraz możliwości wykorzystania wsparcia elektronicznego. W przypadku grup o dużej liczebności przewiduje się formę pisemną, w przypadku grup o niższej liczebności formę sprawdzianu przy stanowisku komputerowym (o ile istnieją takie możliwości infrastrukturalne), również w trybie indywidualnym.

Niezależnie od przyjętej formy realizacji egzaminu prace oceniane są pod kątem stopnia osiągnięcia przez studenta zakładanych efektów kształcenia. W uzasadnionych przypadkach, w porozumieniu z osobami prowadzącymi laboratoria, przewiduje się możliwość zwolnienia z egzaminu, w przypadku uzyskania oceny z zajęć laboratoryjnych, dokumentującej osiągnięcia przez studenta w wysokim stopniu zakładanych efektów kształcenia. Przyjmuje się, że ocena egzaminacyjna będzie nie mniejsza niż ocena laboratoryjna.

Literatura podstawowa:

1. Radosław Sokół, Microsoft Visual Studio 2012. Programowanie w C i C++, Helion, 2013.
2. Mark Summerfield, Biblioteki Qt Zaawansowane programowanie przy użyciu C++, Helion, 2014.
3. Daniluk A., C++Builder Borland Developer Studio 2006. Kompendium programisty, Helion, 2006.

Literatura uzupełniająca:

1. Owczarek A., Microsoft Visual C++ 2008. Praktyczne przykłady, Helion, 2010.
2. Besta P., Visual Studio 2005. Programowanie z Windows API w języku C++, Helion, 2008.

b) Ćwiczenia laboratoryjne

• Treści programowe (tematyka zajęć):

- 1 Struktura aplikacji w środowisku RAD.
- 2 Projekt pierwszej aplikacji, projektowanie warstwy wizualnej.
- 3 Komponenty wizualne, rodzaje, właściwości, funkcje składowe, obsługa zdarzeń.
- 4 Podstawowe komponenty wizualne, przykłady zastosowań, ćwiczenia.
- 5 Złożone komponenty wizualne, przykłady zastosowań, ćwiczenia.
- 6 Dynamiczne tworzenie okien i komponentów wizualnych.
- 7 Komponenty pojemnikowe i zarządzanie kolekcjami elementów, ćwiczenia.
- 8 Grafika i programowanie operacji graficznych.
- 9 Komponenty systemowe, dostępu do baz danych, komponenty sieciowe, przykłady zastosowań, ćwiczenia.
- 10 Implementacja projektu indywidualnego.

• Metody dydaktyczne:

Przedmiot ten realizowany jest w ramach zajęć wykładowych oraz ćwiczeń laboratoryjnych. Wykład stanowi podbudowę ćwiczeń, wprowadzając wszystkie niezbędne zagadnienia. Na tej podstawie realizowane są ćwiczenia, w ramach, których zakłada się realizację bloków tematycznych, obejmujących spójne treściowo przykłady oraz zadania do indywidualnego wykonania.

• Forma i warunki zaliczenia:

Warunkiem zaliczenia przedmiotu jest uzyskanie pozytywnych ocen z elementów, zgodnie z pkt. 8. Ocena jest wypadkową oceny sprawdzianów, prac kontrolnych oraz sprawozdania z projektu.

• Wykaz literatury podstawowej:

1. Wei-Meng Lee, C# 2008. Warsztat programisty, Helion, 2010.
2. Wybrańczyk M., C++Builder 6 i bazy danych, Helion, 2005.

• Wykaz literatury uzupełniającej:

1. Peter Van Roy, Seif Haridi, Programowanie koncepcje techniki i modele, Helion, 2004.
2. Stasiewicz A., C++ Builder. 20 efektywnych programów, Helion, 2002

Zakładane efekty kształcenia

Efekty kształcenia dla modułu: Programowanie w Środowiskach Zintegrowanych		Odniesienie do efektów kształcenia dla kierunku
nr	Opis: student ...	
PSZ_W_01	... zna koncepcję obiektowego, rozumie w jaki sposób wykorzystać techniki programowania obiektowego, zna koncepcję dziedziczenia jedno i wielobazowego, związków całość-część, rozróżnia poprawnie przypadki ich zastosowania, rozumie koncepcję polimorfizmu, zna zasady wykorzystania metod wirtualnych oraz rozumie koncepcje klas abstrakcyjnych.	I1Inż_W01 I1Inż_W02
PSZ_W_02	... rozumie koncepcję programowania sterowanego zdarzeniami w środowiskach, rozróżnia podstawowe komponenty GUI oraz zna zasady ich wykorzystania, zna zasady tworzenia, wie jak programować operacje graficzne i jak wykorzystywać grafikę w aplikacjach. Zna zasady tworzenia GUI.	I1Inż_W02 I1Inż_W03
PSZ_W_03	... zna i rozumie metody wykorzystania zintegrowanych środowisk programistycznych do tworzenia aplikacji klasy desktop i WWW.	I1Inż_W03 I1Inż_W04
PSZ_U_01	... potrafi programować z wykorzystaniem podejścia obiektowego, potrafi stosować dziedziczenie jedno i wielobazowe, związki całość-część, potrafi poprawnie je stosować, potrafi wykorzystywać polimorfizm, stosować metody wirtualne oraz potrafi umiejętnie budować klasy abstrakcyjne i bazujące na nich hierarchie klas.	I1Inż_U01 I1Inż_U02
PSZ_U_02	... potrafi budować aplikacje GUI, potrafi dobierać odpowiednie dobierać odpowiednie komponenty graficzne i kreatywnie je stosować. Student stosuje programowanie sterowane zdarzeniami, definiuje procedury obsługi zdarzeń. Poprawnie programuje podstawowe operacje graficzne, wykorzystuje elementy multimedialne	I1Inż_U02 I1Inż_U03
PSZ_U_03	... potrafi wykorzystywać zintegrowane środowiska programistyczne do projektowania, tworzenia, testowania i uruchamiania aplikacji, optymalizacji jej działania, organizacji pracy grupowej i wersjonowania kodu.	I1Inż_U07 I1Inż_U09
PSZ_K_01	... posiada kompetencje w zakresie pracy grupowej nad projektem, przejawiające się w umiejętności planowania podzadań, metod ich realizacji oraz zarządzania współdzielonym kodem aplikacji.	I1Inż_K01 I1Inż_K02

PSZ_K_02	... potrafi tworzyć złożone a ergonomiczne aplikacje GUI dostosowane do wymagań użytkowników. Posiada kompetencje w zakresie współdziałania z użytkownikiem w zakresie ustalania i formułowania wymagań funkcjonalnych i niefunkcjonalnych, w tym wymagań w zakresie warstwy wizualnej aplikacji.	IInż_K03 IInż_K05
----------	---	----------------------

Odniesienie efektów kształcenia do form zajęć i sposób oceny osiągnięcia przez studenta efektów kształcenia

Efekt nr	Forma zajęć			Sposób sprawdzenia osiągnięcia efektu
	wykład	ćwiczenia	laboratorium	
PSZ_W_01	x			Egzamin
PSZ_W_02	x		x	Egzamin
PSZ_W_03	x		x	Egzamin
PSZ_U_01			x	Praca kontrolna
PSZ_U_02			x	Sprawozdanie z projektu
PSZ_U_03			x	Sprawozdanie z projektu
PSZ_K_01			x	Dyskusja + obserwacja pracy
PSZ_K_02			x	Dyskusja

Kryteria uznania osiągnięcia przez studenta efektów kształcenia

Efekt nr	Efekt jest uznawany za osiągnięty, gdy:
PSZ_W_01	Student poprawnie rozwiąże zadanie egzaminacyjne sprawdzające wiedzę o podejściu obiektowym w programowaniu i wykorzystaniu mechanizmów programowania wizualnego.
PSZ_W_02	Student poprawnie rozwiąże zadanie egzaminacyjne sprawdzające wiedzę na temat budowania klas z wykorzystaniem GUI.
PSZ_W_03	Student poprawnie rozwiąże zadanie egzaminacyjne sprawdzające wiedzę na temat wykorzystania dziedziczenia, hermetyzacji i polimorfizmu w aplikacjach z GUI.
PSZ_U_01	Praca kontrolna zawiera poprawny kod w pełni zgodny ze specyfikacją zadania określonego przez prowadzącego — wykorzystanie elementów GUI.
PSZ_U_02	Sprawozdanie zawiera opis kolejnych etapów realizacji projektu — specyfikację wymagań, założenia projektowe, architekturę systemu, kod programu, opis testów — komponenty wizualne.
PSZ_U_03	Sprawozdanie zawiera opis kolejnych etapów realizacji projektu GUI — specyfikację wymagań, założenia projektowe, architekturę systemu, kod programu, opis testów.
Student w trakcie zajęć laboratoryjnych...	
PSZ_K_01	... poszukiwał materiałów źródłowych i kreatywnie z ich korzystał, formułując własne rozwiązania postawionych problemów programistycznych, szanuje prawa autorskie.
PSZ_K_02	... właściwie dobierał metody i techniki rozwiązania problemów, zadawał merytoryczne pytania i rozumiał otrzymane odpowiedzi, czego wynikiem jest rozwiązanie postawionego zadania.