

Object-oriented programming - opis przedmiotu

Informacje ogólne	
Nazwa przedmiotu	Object-oriented programming
Kod przedmiotu	06.9-WM-ZiIP-ANG-D-10_17
Wydział	Wydział Mechaniczny
Kierunek	Management and Production Engineering
Profil	ogólnoakademicki
Rodzaj studiów	drugiego stopnia z tyt. magistra inżyniera
Semestr rozpoczęcia	semestr zimowy 2018/2019

Informacje o przedmiocie	
Semestr	1
Liczba punktów ECTS do zdobycia	3
Typ przedmiotu	obieralny
Język nauczania	angielski
Sylabus opracował	- dr inż. Grzegorz Pająk - dr inż. Iwona Pająk

Formy zajęć					
Forma zajęć	Liczba godzin w semestrze (stacjonarne)	Liczba godzin w tygodniu (stacjonarne)	Liczba godzin w semestrze (niestacjonarne)	Liczba godzin w tygodniu (niestacjonarne)	Forma zaliczenia
Wykład	15	1	-	-	Zaliczenie na ocenę
Ćwiczenia	15	1	-	-	Zaliczenie na ocenę
Projekt	15	1	-	-	Zaliczenie na ocenę

Cel przedmiotu

Familiarisation with the object-oriented approach to system analysis and design, development of the ability to use a modern programming environment, to the extent that the creation of simple elements of the IT system is facilitated.

Wymagania wstępne

Computer skills.

Zakres tematyczny

Lecture

A reminder of the basic elements of the Pascal language, basic information about the object and event programming. RAD type tools and their use on the example of Borland Delphi. Basic elements of the object model. An object as an element of the real world and its model. Object components: attributes and methods. Classes of objects. An object-oriented approach to the description of systems, analysis of the structure and sample operation. Object-event programme model. Execution of a sample programme, using objects available in the VCL library. Object type of elements of the Pascal language. Syntax of the class definition. Definition of fields. Declaration and definition of procedural and functional methods. Arguments of methods passed, by value and reference. The process of creating and deleting objects: constructor and destructor. References to objects and their components. Design, implementation and testing of the sample class.

Exception classes, reporting and capturing exceptions. Using an exception mechanism to handle errors in a designed class. Overloading of methods. Appeals to overloaded methods. The use of overloaded methods to expand class functionality with a transparent interface. Inheritance, base class and child class (subclass). Generalisation and specialisation, examples of class hierarchies. Syntax of the child class definition. The application of inheritance to create classes of own exceptions. Implementation of a sample hierarchy of classes describing a family of geometric figures.

Practical training

A sample programme in the Delphi environment. Type conversion functions, selected components and their properties, references to component properties, input and display of data using components. Assignment instruction, arithmetic operators, standard functions. Using an exception-handling mechanism to detect errors in a designed programme. Class structure, fields and methods. Defining procedural and functional methods. Transmission of parameters by value and variable, examples of applications. Creating and deleting objects: constructors and destructors. Overloading of methods. Design and implementation of sample classes, modelling selected real objects and processes. Creating applications using objects of defined classes. Inheritance, creating a hierarchy of classes. Virtual methods. Design and implementation of classes using the mechanism of inheritance.

Project

Independent implementation of the object-oriented application, including obtaining the necessary information from the literature, designing the required algorithms and user interface and implementing it, using elements of the programming tool discussed.

Metody kształcenia

Lecture: a conventional lecture

Practical training: problem tasks, case analysis, individual work

The project: project carried out individually or in a group

Efekty uczenia się i metody weryfikacji osiągnięcia efektów uczenia się

Opis efektu	Symbole efektów	Metody weryfikacji	Forma zajęć
The student has detailed knowledge of selected issues of Mechanical Engineering, as broadly understood and associated with Production Engineering and computer-aided management.	• K_W06 • K_W09	• bieżąca kontrola na zajęciach • kolokwium	• Wykład • Ćwiczenia
The student is able to obtain information from literature, databases and other sources and is able to integrate, interpret and critically evaluate it, as well as draw conclusions, therefrom, both formulating it and sufficiently justify opinions on it.	• K_U01	• bieżąca kontrola na zajęciach • kolokwium • przygotowanie projektu	• Wykład • Projekt • Ćwiczenia
The student can work individually as well as in a team; he/she is also able to select team members for a specific task and assign tasks to the members and manage a small team.	• K_U03	• bieżąca kontrola na zajęciach	• Projekt • Ćwiczenia
The student is able to interact and work in a group accepting various roles	• K_K03	• bieżąca kontrola na zajęciach • przygotowanie projektu	• Projekt • Ćwiczenia
The student is able to think and act both creatively and entrepreneurially.	• K_K06	• bieżąca kontrola na zajęciach	• Projekt • Ćwiczenia

Warunki zaliczenia

Lecture: a positive result of the assessment *via* a written test

Practical training: a positive final assessment, based on grades from tests and oral answers

Project: preparation and passing the project

Final rating: the arithmetical mean of grades from individual types of classes.

Literatura podstawowa

1. Booch G., Rumbaugh J., Jacobson I., UML przewodnik użytkownika, WNT, Warszawa 2002,
2. Pacheco X., Teixeira S., Delphi 6. Vademecum Profesjonalisty, tom I i II, Helion, Gliwice 2002,
3. Spolsky J., Projektowanie interfejsu użytkownika. Poradnik dla programistów, MIKOM, Warszawa 2001,
4. Weisfeld M., Myślenie obiektowe w programowaniu, Helion, Gliwice, 2010,
5. Wojtuszkiewicz K., Programowanie strukturalne i obiektowe (Tom 2. Programowanie obiektowe), PWN, Warszawa, 2010.

Literatura uzupełniająca

1. Pamuła T., Aplikacje w Delphi. Przykłady, Helion, Gliwice 2011,
2. Pamuła T., Krawiec S., Programowanie strukturalne i obiektowe w Delphi, Wydawnictwo Politechniki Śląskiej, Gliwice 2005,
3. Sadowski T. M., Praktyczny kurs Delphi, Helion, Gliwice 2003.

Uwagi

Zmodyfikowane przez dr inż. Tomasz Belica (ostatnia modyfikacja: 13-09-2018 21:51)

Wygenerowano automatycznie z systemu SylabUZ