



|   |
|---|
| 1. ma rozszerzoną i pogłębioną wiedzę z matematyki przydatną do formułowania i rozwiązywania złożonych zadań informatycznych dotyczących m. in. programowania w logice, formalnej specyfikacji i weryfikacji oprogramowania - [K1st_W1] |
| 2. ma uporządkowaną i podbudowaną teoretycznie wiedzę ogólną w zakresie kluczowych zagadnień informatyki oraz wiedzę szczegółową w zakresie wybranych zagadnień tej dyscypliny - [K1st_W4]  |
| 3. zna podstawowe techniki, metody oraz narzędzi wykorzystywane w procesie rozwiązywania zadań informatycznych, głównie o charakterze inżynierskim - [K1st_W7]  |
| <b>Umiejętności:</b>  |
| 1. potrafi, formułując i rozwiązując zadania informatyczne, zastosować odpowiednio dobrane metody, w tym metody analityczne, symulacyjne lub eksperymentalne - [K1st_U4]  |
| 2. ma umiejętności formułowania algorytmów i ich implementacji z użyciem przynajmniej jednego z popularnych narzędzi - [K1st_U11]   |
| 3. potrafi organizować, współdziałać i pracować w grupie, przyjmując w niej różne role oraz potrafi odpowiednio określić priorytety służące realizacji określonego przez siebie lub innych zadania - [K1st_U18]                         |
| <b>Kompetencje społeczne:</b>   |
| 1. ma świadomość znaczenia wiedzy w rozwiązywaniu problemów inżynierskich oraz zna przykłady i rozumie przyczyny wadliwie działających systemów informatycznych - [K1st_K2]   |

|  |
|--|
| <b>Sposoby sprawdzenia efektów kształcenia</b>   |
| Efekty kształcenia przedstawione wyżej weryfikowane są w następujący sposób:<br>Ocena formująca:<br>- na podstawie odpowiedzi udzielanych odnośnie realizacji zadań w ramach ćwiczeń;<br>Ocena podsumowująca:<br>Sprawdzanie założonych efektów kształcenia realizowane jest przez:<br>- ocenę umiejętności związanych z realizacją zadań laboratoryjnych (w ciągu semestru każdy student ma do napisania pięć programów o stopniowo wzrastającej skali trudności, w tym związanego z utworzeniem komponentu i jego wykorzystaniem),<br>- ocenę wiedzy i umiejętności związanych z realizacją ćwiczeń laboratoryjnych poprzez sprawdziany,<br>- ocenę wiedzy i umiejętności wykazanych na pisemnym kolokwium zaliczeniowym o charakterze problemowym i praktycznym (kolokwium składa się z 6 tematów/zadań o różnej skali trudności i wynikającej stąd różnej punktacji; maksymalnie można uzyskać 20 punktów; na ocenę 3.0 trzeba zdobyć 11 punktów).<br>Studenci, którzy wyróżniająco wykonali programistyczne zadania laboratoryjne oraz uzyskali bardzo dobre wyniki ze sprawdzianów częściowych są zwalniani z kolokwium zaliczeniowego.  |
| <b>Treści programowe</b>   |
| Program przedmiotu obejmuje następujące zagadnienia:<br>- podstawowe pojęcia związane z programowaniem (programowanie, algorytm, program, język programowania, język ukierunkowany maszynowo, rozkaz, język wyższego rzędu, język uniwersalny, język specjalizowany),<br>- przegląd języków programowania (Ada, Algol, asembler, Basic, C, C++, Cobol, Fortran, HTML, Java, Lisp, Logo, Pascal, PHP, PL/1, Prolog),<br>- sieci działań (schematy blokowe) i symbole stosowane w nich,<br>- maksyminy programistyczne (na podstawie książki D. Van Tassela ?Praktyka programowania?),<br>- ogólne zasady programowania zorientowanego obiektowo (dziedziczność, hermetyczność i polimorfizm),<br>- ogólna charakterystyka pakietu Embarcadero Delphi XE2 (z roku 2012; w kolejnych latach będą przedstawiane następne wersje wraz z ich zakupem do laboratoriów),<br>- podstawowe pojęcia związane z konstruowaniem programów w zintegrowanym systemie programowania Delphi (projekt, formatka, komponent, własność, zdarzenie),<br>- posługiwanie się zintegrowanym pakietem programowania Delphi,<br>- przegląd konstrukcji języka Delphi (program, moduł, biblioteka, funkcje i procedury, klasy i obiekty, typy danych, zmienne, instrukcje),<br>- struktura programu i modułu,<br>- podstawowe elementy języka (symbole podstawowe, słowa kluczowe i dyrektywy języka, identyfikatory, liczby, łańcuchy, w tym łańcuchy znaków Unicode, literały logiczne, komentarze i separatory),<br>- typy danych i ich opis (definiowanie typów, typy proste, łańcuchowe, strukturalne, opisujące obiekty, wskaźnikowe, zgodność typów), w tym typy sparametryzowane,<br>- zmienne (deklaracje zmiennych, zmienne indeksowane, rekordowe, obiektowe, dynamiczne, proceduralne, wariantowe, z początkową wartością, nakładanie zmiennych, literały stałe i zmienne),<br>- wyrażenia (rodzaje operatorów i ich priorytet, składnia wyrażenia, wyrażenie stałe),<br>- instrukcje (proste, strukturalne, asemblerowa),<br>- funkcje i procedury (definicje, rodzaje parametrów, przeciążanie, wywoływanie, konwencje wywoływania),<br>- przetwarzanie obiektów (konstruktory i destruktory, metody statyczne, wirtualne, dynamiczne i abstrakcyjne, obsługa wiadomości, własności), |

|  |                      |                            |
|--|----------------------|----------------------------|
| <ul style="list-style-type: none"> <li>- biblioteki łączone dynamicznie (tworzenie bibliotek, statyczne i dynamiczne pobieranie funkcji i procedur),</li> <li>- pakiety,</li> <li>- przetwarzanie plików,</li> <li>- okienka z komunikatami,</li> <li>- tworzenie komponentów VCL i ich wykorzystanie,</li> <li>- wielowątkowość (synchronizacja wątków, priorytety, oczekiwanie na zakończenie),</li> <li>- sprawdzanie obecności poprzedniego egzemplarza programu.</li> </ul> <p>Na zajęciach laboratoryjnych studenci, po zapoznaniu się ze zintegrowanym środowiskiem programowania Delphi, piszą programy wykorzystujące poznane elementy języka.</p> <p>Metody dydaktyczne:</p> <ol style="list-style-type: none"> <li>1. Wykład: prezentacja multimedialna (każdy wykład) oraz prezentacja pisania i wykonywania wybranych programów bezpośrednio w pakiecie Delphi.</li> <li>2. Ćwiczenia laboratoryjne: ćwiczenia praktyczne dotyczące elementów języka Delphi, pisanie programów okienkowych w tym języku.</li> </ol> |                      |                            |
| <p><b>Literatura podstawowa:</b></p> <ol style="list-style-type: none"> <li>1. A. Marciniak, Język programowania Delphi, Wydawnictwo NAKOM, Poznań</li> </ol>  |                      |                            |
| <p><b>Literatura uzupełniająca:</b></p> <ol style="list-style-type: none"> <li>1. X. Pacheco, S. Teixeira, Delphi 7 - vademecum profesjonalisty, tom 1, Wydawnictwo HELION, Gliwice</li> <li>2. M. Cantu, Delphi 7 - praktyka programowania, tom 1 i 2, Wydawnictwo MIKOM, Warszawa</li> <li>3. A. Marciniak, Borland Delphi 5 Professional - Object Pascal, Wydawnictwo NAKOM, Poznań</li> </ol>  |                      |                            |
| <p><b>Bilans nakładu pracy przeciętnego studenta</b></p>   |                      |                            |
| <p><b>Czynność</b></p>   |                      | <p><b>Czas (godz.)</b></p> |
| 1. Udział w zajęciach laboratoryjnych  |                      | 20                         |
| 2. Przygotowanie do ćwiczeń laboratoryjnych  |                      | 10                         |
| 3. Udział w konsultacjach związanych z realizacją procesu kształcenia, w szczególności ćwiczeń laboratoryjnych   |                      | 3                          |
| 4. Napisanie programów, uruchomienie i weryfikacja (czas poza zajęciami laboratoryjnymi)   |                      | 15                         |
| 5. Przygotowanie do sprawdzianów   |                      | 15                         |
| 6. Udział w wykładach  |                      | 12                         |
| 7. Zapoznanie się ze wskazaną literaturą - 300 stron   |                      | 30                         |
| 8. Przygotowanie do zaliczenia wykładów i udział w kolokwium zaliczeniowym (10 + 2 godz.)  |                      | 12                         |
| <p><b>Obciążenie pracą studenta</b></p>  |                      |                            |
| <p><b>forma aktywności</b></p>   | <p><b>godzin</b></p> | <p><b>ECTS</b></p>         |
| Łączny nakład pracy  | 117                  | 5                          |
| Zajęcia wymagające bezpośredniego kontaktu z nauczycielem  | 37                   | 1                          |
| Zajęcia o charakterze praktycznym  | 45                   | 2                          |