*Załącznik nr 1 do Uchwały nr \_\_\_\_\_\_\_ - 2018/2019 z dnia 25 marca 2019 r.*

*w sprawie wytycznych dla tworzenia i zmian programów studiów pierwszego stopnia, drugiego stopnia*

*oraz jednolitych studiów magisterskich rozpoczynających się od roku akademickiego 2019/2020.*

Opis **zajęć (sylabus)**

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Nazwa zajęć: | | Paradygmaty programowania | | | | | | | **ECTS** | | **4** |
| Nazwa zajęć w j. angielskim: | | Programming paradigms | | | | | | | | | |
| Zajęcia dla kierunku studiów: | | **Informatyka** | | | | | | | | | |
|  | |  | | | | | | | | | |
| Język wykładowy: | | polski | | | | Poziom studiów: | | studia I stopnia | | | |
| Forma studiów: | ◻ stacjonarne  ☒ niestacjonarne | Status zajęć: | ☒ podstawowe  ◻ kierunkowe | ☒ obowiązkowe  ◻ do wyboru | | Numer semestru: 5 | | ☒ semestr zimowy ◻ semestr letni | | | |
|  |  | Rok akademicki, od którego obowiązuje opis (rocznik): | | | | 2019/2020 | Numer katalogowy: | **ZIM-IN-1Z-06L-34** | | | |
|  | | | | | | | | | | | |
| Koordynator zajęć: | |  | | | | | | | | | |
| Prowadzący zajęcia: | |  | | | | | | | | | |
| Jednostka realizująca: | |  | | | | | | | | | |
| Jednostka zlecająca: | |  | | | | | | | | | |
| Założenia, cele i opis zajęć: | | Modele programowania zmieniają się co około 10-20 lat. Od programowania proceduralne (Fortran) przeszliśmy do strukturalnego (Pascal/C) i następnie obiektowego (C++/Java). To ostatnie jest obecnie dominującym paradygmatem i poświęcony mu jest odrębny przedmiot w toku studiów. Model ten pozwala jednak na obiekty mutowalne i efekty uboczne co czyni go niezbyt dobrze dostosowanym do programowania nowoczesnych aplikacji współbieżnych. Student powinien zatem znać także inne paradygmaty i wspierające je języki. Niniejszy przedmiot stanowi przegląd pozostałych znanych paradygmatów: funkcyjnego, logicznego oraz prototypowego.  Podczas zajęć przedstawimy trzy języki programowania:   * Prolog – język logiczny pozwalający w prosty sposób atakować niektóre zadania programistyczne * Ruby – język obiektowy z dynamiczną typizacją i wsparciem dla metaprogramowania * Scala – język ma maszynę wirtualną Java łączący paradygmat funkcyjny z obiektowym | | | | | | | | | |
| Formy dydaktyczne, liczba godzin: | | 1. wykład; liczba godzin 30; 2. laboratorium; liczba godzin 30; | | | | | | | | | |
| Metody dydaktyczne: | | dyskusja, projekt, rozwiązywanie problemu, studium przypadku | | | | | | | | | |
| Wymagania formalne  i założenia wstępne: | | Student potrafi biegle programować przynajmniej w jednym języku wysokiego poziomu i zna dominujący obecnie paradygmat obiektowy | | | | | | | | | |
| Efekty uczenia się: | | Wiedza:  1 – zna popularne paradygmaty programowania (obiektowy, prototypowy, funkcyjny i logiczny) oraz języki je wspierające (Ruby, Prolog, Scala)  2 – rozumie jak działają różne narzędzia komunikacji programisty z językiem: konsola interaktywna, kompilator, maszyna wirtualna  3 – rozumie jak działają różne konstrukcje dla obsługi współbieżności (aktory, futury, pamięć transakcyjna) | | | Umiejętności:  1 – wie czym jest metaprogramowanie i potrafi używać go do tworzenia aplikacji internetowych opartych o bazę danych  2 – posiada wiedzę na temat modeli typizacji: statycznej i dynamicznej oraz silnej i słabej i umiejętnie wykorzystuje ją do rozwiązywania zagadnień programistycznych  3 – zna i potrafi stosować w praktyce podstawowe konstrukcje decyzyjne i zasadnicze struktury danych oraz unikalne cechy prezentowanych języków (dopasowanie wzorców, unifikacja, listy składane, monady) | | | | | Kompetencje: | |
| Sposób weryfikacji efektów uczenia się: | | Praca na ćwiczeniach | | | | | | | | | |
| Forma dokumentacji osiągniętych efektów uczenia się: | | Oddawanie zadania elektroniczne przechowywane na platformie Moodle  Testy elektroniczne archiwizowane na platformie Moodle | | | | | | | | | |
| Elementy i wagi mające wpływ  na ocenę końcową: | | **Ćwiczenia laboratoryjne – 33%, zadania domowe – 33%, kolokwia pisemne – 34%** | | | | | | | | | |
| Miejsce realizacji zajęć: | | Wykład – sala audytoryjna, ćwiczenia laboratoryjne – laboratorium komputerowe | | | | | | | | | |
| Literatura podstawowa i uzupełniająca:   * Bruce A. Tate: Siedem języków w siedem tygodni. Praktyczny przewodnik nauki języków programowania , Helion (2011) | | | | | | | | | | | |
| UWAGI  Ocena dostateczna od 50% punktów, potem co 10% zmienia się o pół stopnia. | | | | | | | | | | | |

Wskaźniki ilościowe charakteryzujące moduł/przedmiot:

|  |  |
| --- | --- |
| Szacunkowa sumaryczna liczba godzin pracy studenta (kontaktowych i pracy własnej) niezbędna dla osiągnięcia zakładanych dla zajęć efektów uczenia się - na tej podstawie należy wypełnić pole ECTS: | **100 h** |
| Łączna liczba punktów ECTS, którą student uzyskuje na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich lub innych osób prowadzących zajęcia: | **2 ECTS** |

Tabela zgodności kierunkowych efektów uczenia się z efektami przedmiotu:

|  |  |  |  |
| --- | --- | --- | --- |
| kategoria efektu | Efekty uczenia się dla zajęć: | Odniesienie do efektów dla programu studiów dla kierunku | Oddziaływanie zajęć na efekt kierunkowy\*) |
| Wiedza 1 | zna popularne paradygmaty programowania (obiektowy, prototypowy, funkcyjny i logiczny) oraz języki je wspierające (Ruby, Prolog, Scala) | K\_W06 / P6S\_WG | 1 |
| Wiedza 2 | rozumie jak działają różne narzędzia komunikacji programisty z językiem: konsola interaktywna, kompilator, maszyna wirtualna | K\_W07 / P6S\_WG | 2 |
| Wiedza 3 | rozumie jak działają różne konstrukcje dla obsługi współbieżności (aktory, futury, pamięć transakcyjna) | K\_W17 / P6S\_WG | 3 |
| Umiejętności 1 | wie czym jest metaprogramowanie i potrafi używać go do tworzenia aplikacji internetowych opartych o bazę danych | K\_U25 / P6S\_UW | 2 |
| Umiejętności 2 | posiada wiedzę na temat modeli typizacji: statycznej i dynamicznej oraz silnej i słabej i umiejętnie wykorzystuje ją do rozwiązywania zagadnień programistycznych | K\_U21 / P6S\_UW | 3 |
| Umiejętności 3 | zna i potrafi stosować w praktyce podstawowe konstrukcje decyzyjne i zasadnicze struktury danych oraz unikalne cechy prezentowanych języków (dopasowanie wzorców, unifikacja, listy składane, monady) | K\_U11 / P6S\_UK | 1 |

\*)

3 – zaawansowany i szczegółowy,

2 – znaczący,

1 – podstawowy,