

Techniki modelowania programów - opis przedmiotu

| Informacje ogólne | |
|---------------------|---|
| Nazwa przedmiotu | Techniki modelowania programów |
| Kod przedmiotu | 11.3-WI-INF-D-TMP |
| Wydział | Wydział Informatyki, Elektrotechniki i Automatyki |
| Kierunek | Informatyka |
| Profil | ogólnoakademicki |
| Rodzaj studiów | drugiego stopnia z tyt. magistra inżyniera |
| Semestr rozpoczęcia | semestr zimowy 2021/2022 |

| Informacje o przedmiocie | |
|---------------------------------|----------------------------|
| Semestr | 1 |
| Liczba punktów ECTS do zdobycia | 5 |
| Typ przedmiotu | obowiązkowy |
| Język nauczania | polski |
| Sylabus opracował | • dr inż. Grzegorz Bazydło |

| 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 | 30 | 2 | 18 | 1,2 | Zaliczenie na ocenę |
| Laboratorium | 30 | 2 | 18 | 1,2 | Zaliczenie na ocenę |

Cel przedmiotu

- Zapoznanie studenta z podstawami inżynierii oprogramowania oraz sposobami modelowania programów.
- Ukształtowanie umiejętności w zakresie modelowania procesów biznesowych.
- Zrozumienie przez studenta zasad modelowania obiektowego.
- Ukształtowanie umiejętności w zakresie modelowania z wykorzystaniem języka UML.

Wymagania wstępne

Programowanie obiektowe

Zakres tematyczny

- Elementy inżynierii oprogramowania. Tworzenie oprogramowania. Kryzys oprogramowania i sposoby przeciwdziałania.
- Modelowanie pojęciowe. Rola modelowania w projektowaniu oprogramowania. Rys historyczny współczesnych technik modelowania.
- Analiza biznesowa. Modelowanie procesów biznesowych w notacji BPMN. Biznesowe przypadki użycia. Tworzenie modelu oprogramowania na podstawie modelu BPMN.
- Język UML. Geneza, definicja i cele powstania UML. Diagramy języka UML. Charakterystyka poszczególnych diagramów UML.
- Wytwarzanie oprogramowania oparte na modelach (Model-Driven Development). Architektura MDA (Model-Driven Architecture).
- Analiza, specyfikacja i dokumentowanie wymagań. Modelowanie przypadków użycia. Projekt architektury rozwiązania.
- Metodyki zwinne. Cykl życia oprogramowania.
- Podstawowe pojęcia obiektowości (np. klasy, dziedziczenie, generalizacja, specjalizacja, polimorfizm) i powiązania między obiektami. Tworzenie modelu systemu.
- Modelowanie interfejsu użytkownika.

Metody kształcenia

Wykład: wykład konwencjonalny/tradycyjny.

Laboratorium: ćwiczenia laboratoryjne.

Efekty uczenia się i metody weryfikacji osiągnięcia efektów uczenia się

| Opis efektu | Symbole efektów | Metody weryfikacji | Forma zajęć |
|-------------|-----------------|--------------------|-------------|
|-------------|-----------------|--------------------|-------------|

| Opis efektu | Symbole efektów | Metody weryfikacji | Forma zajęć |
|---|--|---|--|
| Zna języki i techniki modelowania programów oraz procesów biznesowych. | <ul style="list-style-type: none"> • K_W07 • K_U12 | <ul style="list-style-type: none"> • test | <ul style="list-style-type: none"> • Wykład |
| Potrafi modelować oprogramowanie używając odpowiednich języków modelowania. | <ul style="list-style-type: none"> • K_U12 • K_U14 • K_K03 • K_K04 | <ul style="list-style-type: none"> • bieżąca kontrola na zajęciach | <ul style="list-style-type: none"> • Laboratorium |
| Zna podstawy języka UML, najważniejsze rodzaje diagramów UML i ich zastosowanie. | <ul style="list-style-type: none"> • K_W07 | <ul style="list-style-type: none"> • test | <ul style="list-style-type: none"> • Wykład |
| Rozumie potrzebę modelowania oprogramowania w celu ułatwienia jego projektowania oraz zwiększenia jego wiarygodności. | <ul style="list-style-type: none"> • K_U12 • K_K01 • K_K04 | <ul style="list-style-type: none"> • test | <ul style="list-style-type: none"> • Wykład |
| Zna podstawy programowania obiektowego i potrafi projektować programy, używając obiektowego paradygmatu. | <ul style="list-style-type: none"> • K_W07 • K_U12 | <ul style="list-style-type: none"> • bieżąca kontrola na zajęciach | <ul style="list-style-type: none"> • Laboratorium |

Warunki zaliczenia

Wykład: warunkiem zaliczenia jest uzyskanie pozytywnych ocen z kolokwiów pisemnych przeprowadzonych co najmniej raz w semestrze.

Laboratorium: warunkiem zaliczenia jest uzyskanie pozytywnych ocen ze wszystkich ćwiczeń laboratoryjnych przewidzianych do realizacji w ramach programu laboratorium.

Składowe oceny końcowej: wykład: 50% + laboratorium: 50%.

Literatura podstawowa

1. Sacha K.: Inżynieria oprogramowania, Helion, Gliwice, 2010.
2. Grady B., Rumbaugh J., Jacobson I.: UML przewodnik użytkownika, Wydawnictwa Naukowo-Techniczne, Warszawa, 2002.
3. Dąbrowski W.: Modelowanie systemów informatycznych w języku UML 2.1 w praktyce, Wydawnictwo Naukowe PWN, Warszawa, 2007.
4. Wrycza S.: Język UML 2.0 w modelowaniu systemów informatycznych, Helion, Gliwice, 2006.
5. Drejewicz S.: Zrozumieć BPMN. Modelowanie procesów biznesowych, Helion, Gliwice, 2012.
6. Rasmusson J.: Zwinny samuraj. Jak programują mistrzowie zwinności, Helion, Gliwice, 2012.

Literatura uzupełniająca

1. Sommerville I.: Inżynieria oprogramowania, Wydawnictwa Naukowo-Techniczne, Warszawa, 2003.
2. Graessle P., Baumann H., Baumann P.: UML 2.0 w akcji. Przewodnik oparty na projektach, Helion, Gliwice, 2006.
3. Piotrowski M.: Notacja modelowania procesów biznesowych – podstawy, BTC, Legionowo, 2007.
4. Brookes F. P.: Mityczny osobomiesiąc. Eseje o inżynierii oprogramowania, Wydawnictwa Naukowo-Techniczne, Warszawa, 2000.
5. Osterwalder A., Pigneur Y.: Tworzenie modeli biznesowych. Podręcznik wizjonera, Helion, Gliwice, 2012.
6. Martin R. C.: Zwinne wytwarzanie oprogramowania. Najlepsze zasady, wzorce i praktyki, Helion, Gliwice, 2014.

Uwagi

Zmodyfikowane przez dr inż. Grzegorz Bazydło (ostatnia modyfikacja: 30-04-2021 14:33)

Wygenerowano automatycznie z systemu SyllabUZ