1. Nazwa jednostki prowadzącej kierunek: Polsko-Japońska Akademia Technik Komputerowych
Zamiejscowy Wydział Informatyki w Gdańsku

2. Nazwa kierunku: Informatyka

3. Rodzaj studiów: studia inżynierskie niestacjonarne

4. Profil: ogólnoakademicki

5. Nazwa przedmiotu: Projektowanie systemów informacyjnych

6. Mnemonik: PRI

7. Semestr nauczania: 4

8. Język wykładowy: polski

9. Cel przedmiotu:

Celem przedmiotu jest zapoznanie słuchaczy z technikami, metodyką wytwarzania oprogramowania systemów informacyjnych w oparciu o pojęcia obiektowości. Wprowadzenie stanowi motywacja dla systematycznego wytwarzania oprogramowania, podstawowe pojęcia i założenia inżynierii oprogramowania oraz zasady identyfikacji i specyfikacji wymagań. W dalszym ciągu wykład omawia genezę i zasady obiektowości oraz główne pojęcia paradygmatu obiektowego. Następnie wykład wprowadza w język do modelowania UML (Unified Modeling Language). Omawiane są podstawowe cele i założenia UML, model przypadków użycia, diagram klas i model obiektowy, modele opisu dynamiki oraz diagramy implementacyjne. Ostatnie wykłady są poświęcone metodom odwzorowania schematu obiektowego na schemat relacyjny oraz procesowi analizy i projektowania obiektowego.

10. Forma i sposób zaliczenia oraz kryteria oceny:

A. Sposób zaliczenia:[1]

zaliczenie z oceną

B. Forma zaliczenia:[1]

zaliczenie ustne/kolokwium
ustalenie oceny zaliczeniowej na podstawie ocen cząstkowych otrzymanych w trakcie trwania semestru

C. Zasady oceniania:

Ocenę stanowi 50% kolokwium oraz 50% oceny zaliczeniowej na podstawie ocen otrzymanych w trakcie trwania semestru

11. Formy zajęć, sposób ich realizacji i przypisana im liczba godzin

A. Formy zajęć:[1]

wykład
laboratorium

B. Sposób realizacji zajęć:[1]

zajęcia w salach dydaktycznych PJATK

C. Liczba godzin: [1]

wykład - 16
laboratorium - 16

12. Liczba punktów ECTS: 4

13. Określenie przedmiotów poprzedzających oraz wymagań wstępnych:

A. Przedmioty poprzedzające:

- Programowanie obiektowe
- Wstęp do informatyki i architektura komputerów
- Relacyjne bazy danych

B. Wymagania wstępne:

- Umiejętność programowania imperatywnego
- Wiedza o organizacji systemu komputerowego i architekturze warstwowej
- Ogólna znajomość problematyki relacyjnych baz danych

[1] Pozostawić właściwe

14. Treści programowe:		
Nr tyg.	**Wykład**	**Laboratorium[2]**
1	Wprowadzenie, motywacje dla systematycznego wytwarzania oprogramowania. Pojęcia i metody inżynierii oprogramowania. Cykl życia produktu informatycznego. Narzędzia CASE; rodzaje i ocena narzędzi wspomagających. Faza przedprojektowa i planowanie projektu; Dokument Założeń Wstępnych (DZW)	Dobór tematów Przygotowanie Dokumentu Założeń Wstępnych
2	Inżynieria wymagań; pojęcie wymagania; kategorie i przykłady wymagań. Specyfikacja wymagań i dokument Specyfikacji Wymagań Systemowych (SWS). Praktyczne przykłady konstrukcji SWS	Opracowanie dokumentu SWS w oparciu o DZW (1)
3	Model przypadków użycia. Praktyka modelowania i specyfikacji przypadków użycia. Techniki konceptualnej specyfikacji przypadków użycia.	Opracowanie dokumentu SWS w oparciu o DZW (2)
4	Modelowanie dynamiki systemu w języku UML – wprowadzenie. Techniki modelowania interakcji w systemie. Diagramy sekwencji UML jako uniwersalna technika modelowania interakcji pomiędzy składowymi systemu.	Narzędzie CASE (Enterprise Architect) – obsługa Konstrukcja modelu przypadków użycia w oparciu o specyfikację SWS. Konceptualna specyfikacja przypadków użycia z wykorzystaniem narzędzia Enterprise Architect – scenariusze ustrukturyzowane oraz generowanie konceptualnych diagramów czynności.
5	Specyfikowanie przypadków użycia z wykorzystaniem diagramów czynności	Modele behawioralne: diagramy sekwencji oraz czynności w zależności od specyfiki przypadku użycia (1)
6	Modelowanie struktury systemu: obiekty, klasy, właściwości klasy, dziedziczenie i wielodziedziczenie, polimorfizm, agregacja. Wyrażanie struktury systemu w języku UML	Modele behawioralne: diagramy sekwencji oraz czynności w zależności od specyfiki przypadku użycia (2)
7	Przekształcanie modeli obiektowych w architekturę, kod, interfejs i w trwałe struktury bazodanowe	Konstrukcja modelu strukturalnego – diagram klas dla specyfikowanego systemu. Przejście od modeli do kodu, reprezentacji bazodanowej i interfejsu

[2] Niepotrzebne skreślić

8	Kolokwium	Omówienie jakości dostarczonych komponentów projektowych oraz zaliczenie przedmiotu.

15. Wykaz literatury

A. Literatura podstawowa:

Wrycza S., Marcinkowski B., Wyrzykowski K.: Język UML 2.0 w modelowaniu systemów informatycznych. Helion, 2006.

Szejko S. (red): Metody wytwarzania oprogramowania. MIKOM, 2002.

B. Literatura uzupełniająca:

- Graessle P., UML 2.0 w akcji, Helion, 2010
- Wrycza S., Marcinkowski B., Maślankowski J.: UML. Ćwiczenia zaawansowane. Helion, 2012
- Object Management Group: Unified Modeling Language Specification Version 2.5.1, https://www.omg.org/spec/UML/2.5.1

16. Efekty kształcenia

A. Wiedza	• K_W14: ma uporządkowaną wiedzę zagadnień inżynierii oprogramowania, standardów i kształtu cykli wytwórczych oraz ewolucji oprogramowania; zna podstawy zarządzania przedsięwzięciem programistycznym i rozumie problem jakości oprogramowania; rozumie rolę modelowania i ma szczegółową wiedzę o obiektowym wytwarzaniu oprogramowania i notacji UML, zna zasady korzystania z wzorców programowych i standardowych API; ma wiedzę o typowych narzędziach i środowiskach wspomagających; • K_W15: ma uporządkowaną wiedzę obejmującą kluczowe zagadnienia inżynierii wymagań, rozumie potrzebę systematycznego budowania i pielęgnacji specyfikacji wymagań; ma szczegółową wiedzę dotyczącą ich specyfikacji, analizy i modelowania z użyciem dostępnych narzędzi; • K_W17: ma uporządkowaną wiedzę na temat planowania przedsięwzięcia informatycznego, wstępnej oceny ekonomicznej, aspektów społecznych oraz analizy wykonalności
B. Umiejętności	• K_U16: posiada umiejętność specyfikowania, projektowania, implementacji, testowania i debuggowania programów; potrafi korzystać z bibliotek, środowisk programistycznych, integrujących i uruchomieniowych • K_U19: potrafi zaplanować i zrealizować prosty system oprogramowania zgodnie z metodyką obiektową, posługując się wzorcami programowymi, standardami i dobrymi praktykami programistycznymi; ma umiejętność doboru modelu procesu wytwarzania oprogramowania do specyfiki przedsięwzięcia, a także doboru narzędzi wspomagających budowę oprogramowania • K_U20: potrafi zaplanować i przeprowadzić procesy pozyskiwania, analizy, specyfikacji i modelowania wymagań wobec oprogramowania oraz ich pielęgnacji. • K_U21: potrafi dokonać przeglądu projektu oprogramowania i poprawić jego jakość • K_U26: potrafi zaplanować i wytworzyć podstawowe dokumenty związane z realizacją prostego przedsięwzięcia informatycznego, wstępnie ocenić efekty ekonomiczne i społeczne przedsięwzięcia oraz ich wpływ na udziałowców;
C. Kompetencje społeczne	

17. Weryfikacja efektów kształcenia

	K_W14	K_W15	K_W17	K_U16	K_U19	K_U20	K_U21	K_U26
Zaliczenie ustne/Kolokwium	+	+	+	+				+

Projekty (oceny cząstkowe otrzymywane w trakcie semestru)	+	+		+	+	+	+	+

18. Załączniki

19. Osoby prowadzące:

dr Bartosz Marcinkowski
dr inż. Stanisław Szejko
mgr inż. Grzegorz Cysewski

20. Kontakt:

bartosz.marcinkowski@gmail.com
stasz@pjwstk.edu.pl
grechut@pjwstk.edu.pl