**Nazwa przedmiotu:**

Projekt indywidualny

**Koordynator przedmiotu:**

prof. nzw. dr hab. Jacek Mańdziuk

**Status przedmiotu:**

Obowiązkowy

**Poziom kształcenia:**

Studia I stopnia

**Program:**

Informatyka

**Grupa przedmiotów:**

Wspólne

**Kod przedmiotu:**

**Semestr nominalny:**

6 / rok ak. 2012/2013

**Liczba punktów ECTS:**

4

**Liczba godzin pracy studenta związanych z osiągnięciem efektów uczenia się:**

1. godziny kontaktowe 45h, w tym:
2. obecność na zajęciach w laboratorium - 45h
3. przygotowanie specyfikacji funkcjonalnej aplikacji - 5h
4. przygotowanie projektu technicznego aplikacji – 15h
5. napisanie programu, uruchomienie - 30h
6. przetestowanie aplikacji – 5 h
Razem nakład pracy studenta 100h = 4p. ECTS

**Liczba punktów ECTS na zajęciach wymagających bezpośredniego udziału nauczycieli akademickich:**

a. obecność na zajęciach w laboratorium - 45h
Razem: 45h., co odpowiada 2 punktom ECTS.

**Język prowadzenia zajęć:**

polski

**Liczba punktów ECTS, którą student uzyskuje w ramach zajęć o charakterze praktycznym:**

1. obecność na zajęciach w laboratorium - 45h
2. przygotowanie specyfikacji funkcjonalnej aplikacji - 5h
3. przygotowanie projektu technicznego aplikacji – 15h
4. napisanie programu, uruchomienie - 30h
5. przetestowanie aplikacji – 5 h
Razem: 100h, co odpowiada 4 punktom ECTS.

**Formy zajęć i ich wymiar w semestrze:**

|  |  |
| --- | --- |
| Wykład: | 0h |
| Ćwiczenia: | 0h |
| Laboratorium: | 0h |
| Projekt: | 45h |
| Lekcje komputerowe: | 0h |

**Wymagania wstępne:**

Programowanie Obiektowe i równolegle prowadzony przedmiot Programowanie w Środowisku Graficznym

**Limit liczby studentów:**

Bez limitu

**Cel przedmiotu:**

Celem przedmiotu jest praktyczne wykorzystanie umiejętności z zakresu projektowania aplikacji i programowania do stworzenia projektu informatycznego. Każdy student indywidualnie przygotowuje swój projekt.
Przedmiot podzielony jest na kolejne etapy, odpowiadające kolejnym etapom tworzenia projektu informatycznego:
1. Analiza funkcjonalna. .
2. Projekt techniczny.
3. Implementacja.
4. Testowanie
Po ukończeniu kursu studenci powinni potrafić:
przygotować specyfikację funkcjonalną dla projektu informatycznego.
dobrać odpowiednią architekturę aplikacji i narzędzia niezbędne do realizacji projektu na podstawie specyfikacji.
przygotowywać dokumenty związane z kolejnymi etapami powstawania projektu
pisać zgodny z dobrymi zasadami programowania kod aplikacji i sporządzać dokumentację kodu

**Treści kształcenia:**

Studenci przez cały semestr przygotowują przykładową (większą niż na wcześniejszych semestrach) aplikację.
Wyróżnione i oddzielnie oceniane są następujące etapy pracy nad programem:
• dokumentacja wstępna (opis zadania – analiza funkcjonalna, projekt interfejsu użytkownika)
• wstępny projekt techniczny
• implementacja
• dokumentacja wygenerowana ze źródeł programu i instrukcja dla użytkownika
• testowanie
Studenci pracują indywidualnie, ale bardzo ważnym elementem jest zamiana zadań po każdym z powyższych etapów. Tzn. student, który przygotowuje projekt A, implementuje projekt B i testuje projekt C.
Gotowa aplikacja powinna realizować postawione przed nią zadania (zdefiniowane w projekcie), posiadać wygodny interfejs i instrukcję dla użytkownika. Oceniana jest również jakość kodu źródłowego.

**Metody oceny:**

Ocena końcowa uwzględnia oceny z każdego z etapów:
10% - dokumentacja wstępna
25% - wstępny projekt techniczny
50% - implementacja
5% - dokumentacja wygenerowana ze źródeł programu i instrukcja użytkownika
10% - testowanie
 
Dodatkowo, ocena końcowa jest obniżana na skutek opóźnień w oddawaniu kolejnych etapów projektu. W przypadku opóźnienia oddawania etapów "Dokumentacja wstępna" i "Implementacja wraz z dokumentacją wygenerowaną ze źródeł programu i instrukcją użytkownika", każdy tydzień opóźnienia powoduje utratę 10% ogólnej liczby punktów.

**Egzamin:**

nie

**Literatura:**

brak

**Witryna www przedmiotu:**

brak

**Uwagi:**

## Efekty przedmiotowe

### Profil ogólnoakademicki - wiedza

**Efekt W01:**

Ma uporządkowaną wiedzę dotyczącą projektowania i programowania systemów informatycznych

Weryfikacja:

Zaliczenie projektu

**Powiązane efekty kierunkowe:** K\_W06, K\_W07, K\_W12

**Powiązane efekty obszarowe:** T1A\_W03, T1A\_W03, T1A\_W07

**Efekt W02:**

Ma podstawową wiedzę o cyklu życia systemów informatycznych

Weryfikacja:

Zaliczenie projektu

**Powiązane efekty kierunkowe:** K\_W09

**Powiązane efekty obszarowe:** T1A\_W06

### Profil ogólnoakademicki - umiejętności

**Efekt U01:**

Potrafi przygotować specyfikację funkcjonalną i projekt techniczny aplikacji.

Weryfikacja:

Zaliczenie projektu (specyfikacja funkcjonalna, projekt techniczny)

**Powiązane efekty kierunkowe:** K\_U01, K\_U11, K\_U13, K\_U23, K\_U28, K\_U29, K\_U30

**Powiązane efekty obszarowe:** T1A\_U09, T1A\_U09, T1A\_U14, T1A\_U15, T1A\_U16, T1A\_U09, T1A\_U16, T1A\_U14, T1A\_U15, T1A\_U16

**Efekt U02:**

Potrafi korzystać z dostępnej literatury do rozwiązywania problemów.

Weryfikacja:

Zaliczenie projektu

**Powiązane efekty kierunkowe:** K\_U05

**Powiązane efekty obszarowe:** T1A\_U01

**Efekt U03:**

Potrafi zaimplementować aplikację na podstawie projektu technicznego.

Weryfikacja:

Zaliczenie projektu (implementacja)

**Powiązane efekty kierunkowe:** K\_U11, K\_U21, K\_U29

**Powiązane efekty obszarowe:** T1A\_U09, T1A\_U14, T1A\_U15, T1A\_U09, T1A\_U13, T1A\_U14, T1A\_U15

**Efekt U04:**

Potrafi tworzyć przejrzysty, zgodny z dobrym stylem programowania kod.

Weryfikacja:

Zaliczenie projektu (projekt techniczny, implementacja)

**Powiązane efekty kierunkowe:** K\_U11, K\_U30

**Powiązane efekty obszarowe:** T1A\_U09, T1A\_U14, T1A\_U15, T1A\_U16

**Efekt U05:**

Potrafi przetestować aplikację pod kątem spełnienia wymagać funkcjonalnych.

Weryfikacja:

Zaliczenie projektu (testowanie)

**Powiązane efekty kierunkowe:** K\_U21, K\_U30

**Powiązane efekty obszarowe:** T1A\_U09, T1A\_U13, T1A\_U14, T1A\_U16

### Profil ogólnoakademicki - kompetencje społeczne

**Efekt K01:**

Potrafi indywidualnie realizować zadania i zarządzać swoim czasem.

Weryfikacja:

Zaliczenie projektu

**Powiązane efekty kierunkowe:** K\_K05, K\_K06

**Powiązane efekty obszarowe:** T1A\_K03, T1A\_K04, T1A\_K03, T1A\_K04, T1A\_K06

**Efekt K02:**

Rozumie potrzebę samodzielnego poznawania nowych narzędzi i osiągnięć technologicznych, ponieważ stare szybko stają się przestarzałe.

Weryfikacja:

Zaliczenie projektu

**Powiązane efekty kierunkowe:** K\_K01

**Powiązane efekty obszarowe:** T1A\_K01