



## KARTA OPISU PRZEDMIOTU - SYLABUS

Nazwa przedmiotu

Technologie programistyczne

### Przedmiot

Kierunek studiów

Informatyka

Studia w zakresie (specjalność)

Inteligentne technologie informatyczne

Poziom studiów

drugiego stopnia

Forma studiów

stacjonarne

Rok/semestr

1 / 1

Profil studiów

ogólnoakademicki

Język oferowanego przedmiotu

polski

Wymagalność

obligatoryjny

### Liczba godzin

Wykład

30

Laboratoria

Inne (np. online)

Ćwiczenia

Projekty/seminaria

30

### Liczba punktów ECTS

4

### Wykładowcy

Odpowiedzialny za przedmiot/wykładowca:

dr hab. inż. Szymon Wilk

Odpowiedzialny za przedmiot/wykładowca:

dr inż. Bartłomiej Prędko

### Wymagania wstępne

Student rozpoczynający ten przedmiot powinien posiadać wiedzę z zakresu programowania oraz popularnych technologii programistycznych (np. Java) prezentowaną w ramach wcześniejszych przedmiotów.

Student powinien posiadać umiejętność projektowania i implementowania prostych systemów informatycznych oraz pozyskiwania informacji ze wskazanych źródeł. Student powinien rozumieć konieczność poszerzania swoich kompetencji / mieć gotowość do podjęcia współpracy w ramach zespołu.

Student musi prezentować takie postawy jak uczciwość, odpowiedzialność, wytrwałość, ciekawość poznawcza, kreatywność, kultura osobista, szacunek dla innych ludzi, umiejętność pracy grupowej.

### Cel przedmiotu

1. Przekazanie studentom podstawowej wiedzy z informatyki, w zakresie nowoczesnych technologii programistycznych służących do budowy zaawansowanych systemów informatycznych (np. mobilnych, wykorzystujących techniki uczenia maszynowego) oraz w zakresie projektowania i implementacji tego typu systemów.



2. Rozwijanie u studentów umiejętności rozwiązywania problemów związanych z oceną i wyborem odpowiednich technologii programistycznych dostosowanych do specyfiki, z projektowaniem i implementacją zaawansowanych systemów informatycznych oraz z przygotowaniem prezentacji i sprawozdań opisujących wykonane prace.

3. Kształtowanie u studentów umiejętności pracy zespołowej związanej z realizacją projektów programistycznych w grupach.

### Przedmiotowe efekty uczenia się

#### Wiedza

1. Student ma zaawansowaną i pogłębioną wiedzę z zakresu technologii, narzędzi i środowisk programistycznych wykorzystywanych do implementacji systemów informatycznych (w szczególności systemów mobilnych i inteligentnych).

2. Student ma wiedzę o trendach rozwojowych i najistotniejszych nowych osiągnięciach informatyki z zakresu technologii programistycznych.

#### Umiejętności

1. Student potrafi pozyskiwać informacje z zakresu technologii programistycznych z różnych źródeł (publikacje, zasoby internetowe), właściwie je integrować i interpretować.

2. Student potrafi ocenić przydatność wybranych technologii programistycznych do rozwiązania polegającego na budowie systemu informatycznego (całości lub wybranych modułów). w tym dostrzec ograniczenia tych technologii.

3. Student potrafi - zgodnie z zadaną specyfikacją - zaprojektować i zaimplementować system informatyczny używając właściwych technologii i narzędzi, w tym przystosowując do tego celu istniejące lub opracowując nowe narzędzia.

4. Student potrafi przygotować i przedstawić opracowanie w języku polskim lub angielskim, przedstawiające wyniki procesu implementacji systemu informatycznego ze szczególnym uwzględnieniem wykorzystanych technologii.

#### Kompetencje społeczne

1. Student rozumie, że w informatyce wiedza i umiejętności dotyczące technologii programistycznych bardzo szybko stają się przestarzałe.

2. Student rozumie znaczenie wykorzystywania najnowszej wiedzy z zakresu technologii programistycznych w rozwiązywaniu problemów praktycznych.

### Metody weryfikacji efektów uczenia się i kryteria oceny

Efekty uczenia się przedstawione wyżej weryfikowane są w następujący sposób:

Wiedza nabyta w ramach wykładu jest weryfikowana przez:

- 45-minutowe kolokwium realizowane na ostatnim wykładzie i obejmujące 10-15 pytań (testowych i otwartych), różnie punktowanych. Próg zaliczeniowy: 50% punktów. Zagadnienia zaliczeniowe, na podstawie których opracowywane są pytania zostaną przesłane studentom drogą mailową z



wykorzystaniem systemu uczelnianej poczty elektronicznej,

- prezentację przedstawioną na wykładzie. Prezentacje powinny dotyczyć nowatorskich i mniej typowych (np. zorientowanych naukowo) technologii i narzędzi programistycznych. Powinny być one przygotowywane w grupach 1-2 osobowych. Sugerowany czas prezentacji to 30 minut w przypadku jednego autora i 60 minut w przypadku dwóch autorów. Prezentacje będą oceniane w skali od 2.0 do 5.0. Uzyskanie oceny pozytywnej będzie uprawniało do zwolnienia ze kolokwium. Prezentacje przeznaczone są chętnych, a studenci planujący przygotowanie prezentacji powinni zgłosić ten zamiar w ciągu trzech pierwszych tygodni semestru.

Umiejętności nabyte w ramach zajęć laboratoryjnych weryfikowane są poprzez:

- ocenę prezentacji przedstawiających postępy w realizacji projektów oraz prezentacji końcowej wraz z demonstracją zrealizowanego projektu. Każda grupa projektowa zobowiązana jest przygotować trzy prezentacje w ciągu semestru,  
- ocenę końcowego sprawozdania pisemnego opisującego zrealizowany projekt i przygotowywanego częściowo w trakcie zajęć, a częściowo po ich zakończeniu.

Ostateczna ocena z projektu jest wyznaczona jako średnia powyższych ocen.

### Treści programowe

Program wykładu obejmuje następujące zagadnienia (ze względu na szybki dziedzinny rozwój wskazane poniżej technologie i narzędzia ze charakteru przykładowy i mogą zostać zmienione na nowsze odpowiedniki):

1. Historia rozwoju platform mobilnych,
2. Narzędzia, technologie oraz usługi do rozwoju aplikacji mobilnych, w szczególności aplikacji dla platformy iOS,
3. Narzędzia i biblioteki służące do analizy danych i budowy modeli decyzyjnych przy użyciu technik uczenia maszynowego, np. TensorFlow, Microsoft Cognitive Services,
4. Narzędzia i biblioteki służące do optymalizacji i rozwiązywania problemów przeszukiwania z ograniczeniami (ang. constraint satisfaction problems), np. MiniZinc, ECLiPSe,
5. Narzędzia i biblioteki służące do analizy, przetwarzania i przeszukiwania dokumentów tekstowych, np. Apache Lucene, Elasticsearch, Spacy,
6. Narzędzia i biblioteki służące do budowy zaawansowanych środowisk dla przetwarzania danych, w tym danych nieustrukturalizowanych, np. Apache UIMA.

Zajęcia projektowe mają charakter seminaryjny i konsultacyjny. Podczas zajęć studenci w grupach dwuosobowych realizują wybrane przez siebie projekty (w szczególnych przypadkach możliwa jest także praca indywidualna). Celem projektów jest zaprojektowanie i implementacja zaawansowanego narzędzia lub systemu informatycznego przy wykorzystaniu technologii prezentowanych na wykładach. Tematy projektów związane są z charakterem specjalizacji i dotyczą szeroko pojętego podejmowania decyzji i przetwarzania danych. Program laboratorium obejmuje:

1. wybór projektu przez każdą z grup oraz zapoznanie się z niezbędną wiedzą dziedzinową,
2. opracowanie specyfikacji wymagań dla projektu, wybór technologii oraz przygotowanie harmonogramu prac,



3. przygotowanie formalnego projektu realizowanego systemu, jego implementację oraz przeprowadzenie testów lub eksperymentów obliczeniowych,
5. przygotowanie prezentacji oraz sprawozdania końcowego łącznie 3 prezentacje i jedno sprawozdanie),
6. przedstawienie prezentacji oraz demonstrację gotowego projektu.

### Metody dydaktyczne

1. Wykład: prezentacja multimedialna oraz prezentacja ilustrowana przykładami podawanymi na tablicy,
2. Projekt: realizacja projektów informatycznych w grupach dwuosobowych, w tym praca w zespole, pokaz multimedialny, demonstracja.

### Literatura

#### Podstawowa

1. D. Conway, J. Myles White: *Uczenie maszynowe dla programistów*, Helion, 2015.
2. M.A. Lasoff: *Podstawy języka Swift : programowanie aplikacji dla platformy iOS*, Helion, 2016.
3. M. Galloway: *Effective Objective-C 2.0 : 52 specific ways to improve your iOS and OS X programs*, Addison-Wesley, 2013.

#### Uzupełniająca

1. A. Niederliński: *Programowanie w logice z ograniczeniami. Łagodne wprowadzenie dla platformy ECLiPSe*. <http://www.pwlzo.pl/>
2. dokumentacja użytkowa i techniczna wybranych narzędzi i środowisk

### Bilans nakładu pracy przeciętnego studenta

	Godzin	ECTS
Łączny nakład pracy	110	4,0
Zajęcia wymagające bezpośredniego kontaktu z nauczycielem	60	2,0
Praca własna studenta (studia literaturowe, wykonanie projektu i jego dokumentacji, przygotowanie się do zajęć, przygotowanie do kolokwium lub prezentacji) <sup>1</sup>	50	2,0

<sup>1</sup> niepotrzebne skreślić lub dopisać inne czynności