

KARTA PRZEDMIOTU**I. Dane podstawowe**

| | |
|--|---|
| Nazwa przedmiotu | Laboratorium programowania - frameworki aplikacji internetowych |
| Nazwa przedmiotu w języku angielskim | Application development with web frameworks |
| Kierunek studiów | Informatyka |
| Poziom studiów (I, II, jednolite magisterskie) | I stopnia |
| Forma studiów (stacjonarne, niestacjonarne) | Stacjonarne |
| Dyscyplina | Informatyka, informatyka techniczna i telekomunikacja |
| Język wykładowy | Polski |

| | |
|---|-------------------------|
| Koordinator przedmiotu/osoba odpowiedzialna | mgr Krzysztof Buszowski |
|---|-------------------------|

| Forma zajęć (<i>katalog zamknięty ze słownika</i>) | Liczba godzin | Semestr | Punkty ECTS |
|--|---------------|---------|-------------|
| wykład | | | 3+3 |
| konwersatorium | | | |
| ćwiczenia | | | |
| laboratorium | 30+30 | V+VI | |
| warsztaty | | | |
| seminarium | | | |
| proseminarium | | | |
| lektorat | | | |
| praktyki | | | |
| zajęcia terenowe | | | |
| pracownia dyplomowa | | | |
| translatorium | | | |
| wizyta studyjna | | | |

| | |
|-------------------|---|
| Wymagania wstępne | Wstęp do programowania Podstawy informatyki i programowania Programowanie obiektowe |
|-------------------|---|

II. Cele kształcenia dla przedmiotu

| |
|--|
| Wykorzystanie rozproszonego systemu kontroli wersji (GIT) oraz narzędzi zarządzania projektem |
| Omówienie wybranych framework'ów umożliwiających implementację części serwerowej (back-end) |
| Omówienie wybranych framework'ów oraz bibliotek umożliwiających implementację aplikacji internetowej (front-end) |
| Omówienie dobrych praktyk programistycznych |

III. Efekty uczenia się dla przedmiotu wraz z odniesieniem do efektów kierunkowych

| Symbol | Opis efektu przedmiotowego | Odniesienie do efektu kierunkowego |
|------------------------------|---|------------------------------------|
| WIEDZA | | |
| W_01 | Student potrafi stosować rozproszony system kontroli wersji oraz narzędzia zarządzania projektem | K_W01, K_W02, K_W07 |
| W_02 | Student potrafi wykorzystać biblioteki oraz framework'i | K_W01, K_W02, K_W07 |
| UMIEJĘTNOŚCI | | |
| U_01 | Student potrafi stworzyć aplikację internetową utrwalającą swój stan (np. korzystając z relacyjnej bazy danych) | K_U03, K_U17 |
| U_02 | Student potrafi stworzyć aplikację wykorzystując architekturę opartą o REST | K_U03, K_U17 |
| KOMPETENCJE SPOŁECZNE | | |
| K_01 | Ma świadomość ograniczenia swojej wiedzy i umiejętności, rozumie potrzebę dokształcania się i podnoszenia kompetencji zawodowych i osobistych | K_K01 |

IV. Opis przedmiotu/ treści programowe

| |
|---|
| <ol style="list-style-type: none"> 1. Wstęp do zajęć. Opis technologii, które będą wykorzystywane na zajęciach. Wstęp do git'a. 2. Kontynuacja git. Praca z repozytoriami. Podstawowe komendy git. Praca z gałęziami. 3. Wstęp do Java. Krótka o JVM i ekosystemie. Omówienie składni i proste zadania do zrobienia. 4. Zarządzanie zależnościami oraz proces budowania z wykorzystaniem Maven'a. Build lifecycle. Testy (unit + it). 5. Potrzeba zapisywania stanu aplikacji. Zapis do pliku bądź bazy danych. Połączenie aplikacji z bazą danych. 6. Połączenie aplikacji z bazą danych. Zagrożenia związane z implementacją rozwiązań (SQL Injection, XSS) 7. Protokół HTTP/2. Jak wygląda żądanie oraz odpowiedź HTTP. Implementacja prostego serwera HTTP w oparciu o framework Spark. 8. REST. Co to jest i po co tego używać. W jaki sposób projektować API. Zabezpieczanie API (uwierzytelnianie użytkownika, CSRF) 9. Wprowadzenie do Spring + Spring MVC. Implementacja serwera HTTP za pomocą kontenera używając adnotacji. Testy integracyjne. 10. Integracja z zewnętrznym serwisem za pomocą REST. 11. WebSocket. Co to jest i po co tego używać. Czym różni się od HTTP/REST. Testy integracyjne. 12. Wstęp do JavaScript. Omówienie składni i proste zadania do zrobienia. 13. Implementacja strony w oparciu o AngularJS z wykorzystaniem API przygotowanego na wcześniejszych zajęciach. 14. Uruchomienie aplikacji na serwerze HTTP z node.js. |
|---|

V. Metody realizacji i weryfikacji efektów uczenia się

| Symbol efektu | Metody dydaktyczne <i>(lista wyboru)</i> | Metody weryfikacji <i>(lista wyboru)</i> | Sposoby dokumentacji <i>(lista wyboru)</i> |
|---------------|---|---|---|
| WIEDZA | | | |
| W_01 | Praca pod kierunkiem | Projekt | Karta oceny projektu |

| | | | |
|------------------------------|---------------------------------|---------|----------------------|
| W_02 | Praca pod kierunkiem | Projekt | Karta oceny projektu |
| UMIEJĘTNOŚCI | | | |
| U_01 | Ćwiczenia praktyczne | Projekt | Karta oceny projektu |
| U_02 | Ćwiczenia praktyczne | Projekt | Karta oceny projektu |
| KOMPETENCJE SPOŁECZNE | | | |
| K_01 | Dyskusja, Metoda problemowa PBL | Projekt | Karta oceny projektu |

VI. Kryteria oceny, wagi...

Warunkiem zaliczenia przedmiotu jest obecność na zajęciach oraz przygotowanie projektu zaliczeniowego w wybranej technologii.

Obciążenie pracą studenta

| | |
|--|--------------------|
| Forma aktywności studenta | Liczba godzin |
| Liczba godzin kontaktowych z nauczycielem | 100 (50+50) |
| Liczba godzin indywidualnej pracy studenta | 60 (30+30) |

VII. Literatura

| |
|---|
| Literatura podstawowa |
| Benjamin J. Evans, David Flanagan, "Java w pigułce. Wydanie VI", Helion 2015 Zasoby internetowe (w tym dokumentacja techniczna stosowanych technologii) Materiały przygotowane przez prowadzącego zajęcia |
| Literatura uzupełniająca |
| Benjamin J. Evans, Martijn Verburg, "The Well-Founded Java Developer", Manning 2012 |