



KAPITAŁ LUDZKI
NARODOWA STRATEGIA SPÓJNOŚCI

Projekt współfinansowany przez
Unię Europejską w ramach
Europejskiego Funduszu
Społecznego

UNIA EUROPEJSKA
EUROPEJSKI
FUNDUSZ SPOŁECZNY



| | | | |
|---|-----------------|--|--|
| Nazwa przedmiotu | | Kod ECTS | |
| Testowanie aplikacji Java | | 11.3.0797 | |
| Nazwa jednostki prowadzącej przedmiot | | | |
| Faculty of Mathematics, Physics and Informatics | | | |
| Studia | | | |
| wydział | kierunek | poziom | pierwszego stopnia |
| Wydział Matematyki, Fizyki i Informatyki | Informatyka | forma | stacjonarne |
| | | moduł | informatyka ogólna, tester programista |
| | | specjalnościowy | |
| | | specjalizacja | wszystkie |
| Nazwisko osoby prowadzącej (osób prowadzących) | | | |
| dr Jakub Neumann | | | |
| Formy zajęć, sposób ich realizacji i przypisana im liczba godzin | | Liczba punktów ECTS | |
| Formy zajęć | | 5 | |
| Wykład, Ćw. laboratoryjne | | | |
| Sposób realizacji zajęć | | | |
| zajęcia w sali dydaktycznej | | | |
| Liczba godzin | | | |
| Ćw. laboratoryjne: 45 godz., Wykład: 15 godz. | | | |
| Cykl dydaktyczny | | | |
| 2017/2018 letni | | | |
| Status przedmiotu | | Język wykładowy | |
| obowiązkowy | | polski | |
| Metody dydaktyczne | | Forma i sposób zaliczenia oraz podstawowe kryteria oceny lub wymagania egzaminacyjne | |
| <ul style="list-style-type: none"> - Dyskusja - Metoda projektów (projekt badawczy, wdrożeniowy, praktyczny) - Projektowanie doświadczeń - Wykonywanie doświadczeń - Wykład z kodowaniem na żywo fragmentów programów i eksperymentami - Wykład z prezentacją multimedialną | | Sposób zaliczenia | |
| | | <ul style="list-style-type: none"> - Zaliczenie na ocenę - Egzamin | |
| | | Formy zaliczenia | |
| | | <ul style="list-style-type: none"> - egzamin ustny - wykonanie pracy zaliczeniowej - projekt lub prezentacja - ustalenie oceny zaliczeniowej na podstawie ocen cząstkowych otrzymywanych w trakcie trwania semestru | |
| | | Podstawowe kryteria oceny | |
| | | Ocenie podlega bieżąca praca studenta na zajęciach, jego rozumienie omawianych zagadnień oraz umiejętność praktycznego stosowania omawianych technik i technologii | |
| Sposób weryfikacji założonych efektów kształcenia | | | |
| | | | |

| zszkaldany efekt kształcenia | Dyskusja | Metoda projektów (projekt badawczy, wdrożeniowy, praktyczny) | Projektowanie doświadczeń | Wykonywanie doświadczeń | Wykład z prezentacją multimedialną | Wykład z kodowaniem na żywo fragmentów programów i eksperymentami |
|------------------------------|----------|--|---------------------------|-------------------------|------------------------------------|---|
| Wiedza | | | | | | |
| K_W04 | x | | x | | x | x |
| K_W09 | x | | x | | x | x |
| K_W10 | x | | x | | x | x |
| Umiejętności | | | | | | |
| K_U02 | | x | | x | | |
| K_U05 | | x | | x | | |
| K_U18 | | x | | x | | |
| Kompetencje | | | | | | |
| K_K01 | x | | | | x | |
| K_K02 | x | | | | x | |
| K_K05 | x | | | | x | |

Określenie przedmiotów wprowadzających wraz z wymogami wstępnymi

A. Wymagania formalne

Zaliczony przedmiot poświęcony programowaniu obiektowemu oraz językowi Java.

B. Wymagania wstępne

Dobra znajomość języka Java oraz zasad projektowania i programowania obiektowego

Cele kształcenia

Celem przedmiotu jest zapoznanie studentów z zasadami, pojęciami oraz technikami i technologiami automatycznego testowania aplikacji w języku Java.

Treści programowe

1. Testowanie jednostkowe w języku Java. Technologie JUnit, Hamcrest
2. Zasady Test Driven Development, tworzenie scenariuszy testowych
3. Zapewnienie niezależności testowanego kodu, atrapy, biblioteki Mockito, Easy Mock
4. Testowanie aplikacji wykorzystujących relacyjne bazy danych, biblioteka DBUnit
5. Testy integracyjne. Wsparcie frameworka Spring dla testów integracyjnych.
6. Testy automatyczne GUI aplikacji webowych. Technologia Selenium.
7. Testowanie usług webowych aplikacji opartych o REST
8. Testowanie wydajności, narzędzie JMeter
9. Zasady Behavior Driven Development oraz tworzenie specyfikacji na potrzeby testów

Wykaz literatury

1. <http://junit.org/>
2. <https://code.google.com/p/mockito/>
3. <http://easymock.org/>
4. <http://www.dbunit.org/>
5. <http://docs.seleniumhq.org/>
6. <https://spring.io/>
7. <http://jmeter.apache.org/>
8. <http://jnarior.org/>
9. Liczne artykuły, blogi uaktualniane na bieżąco

Efekty kształcenia

(obszarowe i kierunkowe)

K_W04 zna podstawowe konstrukcje programistyczne oraz struktury danych
K_W09 ma wiedzę na temat inżynierii oprogramowania, projektowania, wzorców projektowych, wykorzystania API, narzędzi i środowisk wytwarzania oprogramowania, cyklu życia projektu informatycznego, specyfikacji

Wiedza

- K_W04 zna podstawowe konstrukcje programistyczne oraz struktury danych szczególnie w zakresie tworzenia testów jednostkowych, mocków i atrap
- K_W09 ma wiedzę na temat inżynierii oprogramowania, projektowania, wzorców projektowych, wykorzystania API, narzędzi i środowisk wytwarzania oprogramowania, cyklu życia projektu informatycznego, specyfikacji oprogramowania, walidacji i weryfikacji, utrzymywania oprogramowania ze

| | |
|---|---|
| <p>oprogramowania, walidacji i weryfikacji, utrzymywania oprogramowania</p> <p>K_W10 ma wiedzę na temat technologii sieciowych, w tym podstawowych protokołów komunikacyjnych, bezpieczeństwa i budowy aplikacji sieciowych</p> <p>K_U02 potrafi pozyskiwać informacje z literatury, baz wiedzy, Internetu oraz innych wiarygodnych źródeł, integrować je, dokonywać ich interpretacji oraz wyciągać wnioski i formułować opinie</p> <p>K_U05 potrafi pisać, uruchamiać i testować programy w wybranym środowisku programistycznym</p> <p>K_U18 potrafi - zgodnie z zadaną specyfikacją - zaprojektować oraz zrealizować prosty system informatyczny</p> <p>K_K01 zna ograniczenia własnej wiedzy i rozumie potrzebę dalszego kształcenia</p> <p>K_K02 potrafi precyzyjnie formułować pytania, służące pogłębieniu własnego zrozumienia danego tematu lub odnalezieniu brakujących elementów rozumowania</p> <p>K_K05 potrafi samodzielnie wyszukiwać informacje w literaturze, także w językach obcych</p> | <p>szczególnym uwzględnieniem procesów testowania, jednostkowego i integracyjnego</p> <ul style="list-style-type: none"> • K_W10 ma wiedzę na temat technologii sieciowych, w tym podstawowych protokołów komunikacyjnych, bezpieczeństwa i budowy aplikacji sieciowych ze szczególnym uwzględnieniem aplikacji korzystających z protokołu HTTP a także sposobów ich testowania |
| | <p>Umiejętności</p> <ul style="list-style-type: none"> • K_U02 potrafi pozyskiwać informacje z literatury, baz wiedzy, Internetu oraz innych wiarygodnych źródeł, integrować je, dokonywać ich interpretacji oraz wyciągać wnioski i formułować opinie szczególnie w odniesieniu do omawianych na wykładzie technologii związanych z testowaniem • K_U05 potrafi pisać, uruchamiać i testować programy w wybranym środowisku programistycznym • K_U18 potrafi - zgodnie z zadaną specyfikacją - zaprojektować oraz zrealizować prosty system informatyczny a także testować poszczególne jego komponenty, także w izolacji od pozostałych komponentów |
| | <p>Kompetencje społeczne (postawy)</p> <ul style="list-style-type: none"> • K_K01 zna ograniczenia własnej wiedzy i rozumie potrzebę dalszego kształcenia szczególnie w odniesieniu do rozwoju technologii testowania • K_K02 potrafi precyzyjnie formułować pytania, służące pogłębieniu własnego zrozumienia danego tematu lub odnalezieniu brakujących elementów rozumowania szczególnie w odniesieniu do procesu testowania • K_K05 potrafi samodzielnie wyszukiwać informacje w literaturze, także w językach obcych szczególnie w odniesieniu do dokumentacji technicznej omawianych technologii |
| <p>Kontakt</p> <p>jakub.neumann@inf.ug.edu.pl</p> | |