

KARTA PRZEDMIOTU DLA NABORU 2023/2023
FORMA STUDIÓW: STACJONARNA/NIESTACJONARNA

INFORMACJE OGÓLNE

1. Nazwa przedmiotu Automatyzacja testów oprogramowania

2. Nazwa kierunku Informatyka

3. Poziom kształcenia studia drugiego stopnia

4. Liczba punktów ECTS 4

5. Liczba godzin w semestrze

| semestr | W S/NS | ćw | lab/lek S/NS | prj/zp | prk |
|---------|-----------|----|-----------------|--------|-----|
| I | 15/9 | | 30/18 | | |

6. Język wykładowy polski

7. Wykładowca dr inż. Marcin Klimek

INFORMACJE SZCZEGÓŁOWE

8. Wymagania wstępne

- Posiadanie podstawowej wiedzy programistycznej i znajomości środowiska programistycznego.
- Znajomość podstawowych zagadnień związanych z testowaniem oprogramowania w tym z testowania manualnego.

9. Cele przedmiotu

C1 nabycie umiejętności automatyzacji testów oprogramowania

C2 zapoznanie studentów z narzędziami i środowiskami automatyzacji testów

C3 nabycie umiejętności wykorzystania zalet programowania obiektowego oraz refaktoryzacji w celu poprawy jakości oprogramowania i wydajniejszego testowania

10. Efekty uczenia się w zakresie wiedzy, umiejętności i kompetencji społecznych

| | |
|------------------------------------|---|
| Student, który zaliczył przedmiot: | odniesienie do kierunkowych efektów uczenia się |
|------------------------------------|---|

WIEDZA

| | | |
|------|--|--------------|
| EU01 | zna standardy i dobre praktyki wykonywania testów automatycznych | K_W02, K_W03 |
| EU02 | zna narzędzia i zagadnienia związane z technikami automatycznego testowania oprogramowania | K_W03, K_W04 |
| EU03 | rozumie kluczową rolę testowania automatycznego w tym procesie rozwoju i utrzymywania oprogramowania | K_W02, K_W03 |

UMIEJĘTNOŚCI

| | | |
|------|---|--------------|
| EU04 | umie testować oprogramowanie oraz stosuje właściwe testy na odpowiednim etapie procesu wytwórczego oprogramowania | K_U03, K_U04 |
| EU05 | potrafi dobrać odpowiednie narzędzia do przeprowadzenia odpowiedniego typu testów oprogramowania. | K_U03, K_U04 |
| EU06 | potrafi stosować narzędzia testujące w procesach wytwarzania | K_U03, K_U04 |

| | |
|--|---|
| oprogramowania. | |
| KOMPETENCJE SPOŁECZNE | |
| EU07 | rozumie rolę dokumentacji oraz testów w powstawaniu złożonych systemów informatycznych tworzonych zespołowo i umie je tworzyć, rozumie rolę testera i zna konsekwencje niedokładnej pracy |
| EU08 | zna kodeks etyczny i zagadnienia etyczne związane z pracą testera |
| 11. Treści programowe | |
| Forma zajęć – wykłady/ ćwiczenia/laboratoria/zajęcia praktyczne itp. | |
| Wykłady: <ol style="list-style-type: none"> 1. Wprowadzenie do testowania. Testowanie w cyklu życia oprogramowania. 2. Charakterystyka i podział testów oprogramowania. Zastosowania testów w procesie rozwoju oprogramowania: Poziomy testowania oprogramowania. Testy jednostkowe, testy integracyjne wewnętrzne, testy systemowe, testy integracyjne zewnętrzne, testy akceptacyjne itp. 3. Wprowadzenie do automatyzacji testów i jej cele. Architektura testów automatycznych. 4. Raporty i metryki dotyczące automatyzacji testowania. Rola testów automatycznych w tworzeniu oprogramowania. 5. Testy jednostkowe z wykorzystaniem np. XUnit, Mock. Mechanizmy w testach jednostkowych tj. asercje, adnotacje, refleksje oraz przykłady ich użycia. Testy integracyjne. 6. Testowanie interfejsów graficznych z użyciem Cypress, Playwright. 7. Konceptcje Test Driven Development (TDD) oraz Behavior-Driven Development (BDD). 8. Przykłady testów wykorzystujących język naturalny korzystających z Behave. 9. Rola refaktoryzacji w tworzeniu dobrego oprogramowania. 10. Serwery ciągłej integracji w procesie testowania na przykładzie Jenkinsa oraz Dockera. Laboratorium: <ol style="list-style-type: none"> 1. Statyczna inspekcja i przeglądy kodu z użyciem SonarQube: sprawdzanie kodu zgodnie z listą potencjalnych błędów. 2. Projektowanie przypadków testowych w testach czarno skrzynkowych i biało skrzynkowych. 3. Testowanie aplikacji internetowych i systemów rozproszonych. 4. Wykorzystanie narzędzi i frameworków do automatyzacji: XUnit, Selenium, Playwright itp. Testy jednostkowe, oparte na wyjątkach, integracyjne itp. 5. Opracowywanie automatów testowych (funkcjonalnych i wydajnościowych) dla rzeczywistej aplikacji. | |
| 12. Narzędzia/metody dydaktyczne | |
| <ol style="list-style-type: none"> 1. Podczas laboratoriów praca indywidualna studentów: pisanie testów np. jednostkowych. Studenci samodzielnie rozwiązują zadany problem praktyczny, dobierając odpowiednie narzędzia. <ol style="list-style-type: none"> a. dyskusje wokół tematyki poruszanej na wykładzie, b. praktyczne projektowanie przypadków testowych z testami zautomatyzowanymi 2. Laboratoria w części odbywają się przy tablicy: studenci np. analizują różne strategie dotyczące testowania oprogramowania. 3. Wykład prowadzony z wykorzystaniem projektora multimedialnego. Wykłady przygotowane w formie prezentacji PowerPoint. 4. Treści prezentowane na wykładzie są wzbogacone o pokazy praktycznych rozwiązań odnoszących się do prezentowanych zagadnień. | |
| 13. Sposoby oceny (częstkowe, końcowe) | |
| <ol style="list-style-type: none"> 1. Kolokwium przy komputerze: implementacja testów automatycznych, implementacja testów integracyjnych 2. Ocena realizacji projektu zaliczeniowego. Automatyzacja testów interfejsu użytkownika w narzędziach Selenium i Playwright. 3. Ocena ciągła pracy indywidualnej podczas laboratoriów. | |
| 14. Obciążenie pracą studenta | |
| Forma aktywności | liczba godzin S/NS |
| 1. Zajęcia z bezpośrednim udziałem nauczyciela oraz konsultacje | 55/27 |

| | |
|--|-------|
| 2. Nakład pracy studenta | 45/73 |
| suma | 100 |
| liczba punktów ECTS | 4 |
| 15. Literatura | |
| Literatura podstawowa: | |
| 1. Vladimir Khorikov, Testy jednostkowe : zasady, praktyki i wzorce, Gliwice : Wydawnictwo Helion, 2020. | |
| 2. Arnon Axelrod, Automatyzacja testów. Kompletny przewodnik dla testerów oprogramowania. PWN, Warszawa, 2020. | |
| 3. Adam Roman, Testowanie i jakość oprogramowania. Modele, techniki, narzędzia, PWN 2015. | |
| 4. Dokumentacja techniczna omawianych technologii, tutoriale, kursy na platformach e-learningowych | |
| Literatura uzupełniająca: | |
| 1. Roy Osheroove, Testy jednostkowe - świat niezawodnych aplikacji , Helion 2014 | |
| 2. Glenford Myers i in., Sztuka testowania oprogramowania, 2005 | |
| 3. Black Mitchell, Advanced Software Testing, Helion 2010 | |
| 4. Mark Fewster, Dorothy Graham, Experiences of Test Automation: Case Studies of Software Test Automation, Addison-Wesley 2012 | |
| 16. Formy oceny – szczegóły | |
| Warunki uzyskania zaliczenia przedmiotu: zajęcia kończą się zaliczeniem z oceną. Sposób weryfikacji efektów uczenia się: Ocena stopnia osiągniętych przez studenta efektów uczenia się następuje wg poniższych kryteriów: 5.0 – zakładany efekt uczenia się został osiągnięty bez zastrzeżeń 4.5 – zakładany efekt uczenia się został osiągnięty z pojedynczymi brakami/błędami 4.0 – zakładany efekt uczenia się został osiągnięty z nielicznymi brakami/błędami 3.5 – zakładany efekt uczenia się został osiągnięty z wieloma brakami/błędami 3.0 – zakładany efekt kształcenia został osiągnięty z licznymi i istotnymi brakami/błędami (minimalnie wymagany poziom osiągnięcia efektu) 2.0 – zakładany efekt uczenia się nie został osiągnięty | |
| 17. Inne przydatne informacje o przedmiocie | |
| 1. Bezpośrednich informacji o problematyce zajęć i treściach programowych udziela Prowadzący w trakcie zajęć i podczas konsultacji | |
| 2. Zajęcia odbywać się będą w Akademii Białskiej im. Jana Pawła II lub na platformie e-learningowej | |
| 3. Zajęcia odbywać się będą zgodnie z aktualnym planem zajęć | |
| 4. Konsultacje odbywać się będą zgodnie z obowiązującym terminarzem | |