

Regulamin uczestnictwa w kursie dodatkowym “Praktyczne Aspekty Rozwoju Oprogramowania”

wykłady organizowane w semestrze letnim roku akademickiego 2021/22
przez Nokia oraz Wydział Informatyki i Telekomunikacji Politechniki Wrocławskiej

1 Zakres wykładów i laboratoriów

Nazwa w języku polskim: Praktyczne Aspekty Rozwoju Oprogramowania
Nazwa w języku angielskim: Practical Aspects of Software Development
Forma prowadzenia zajęć: zdalna

Terminy zajęć na Politechnice Wrocławskiej (platforma MS Teams):

Wykłady	środy 17:05-19:40 (w dniach: 02.03., 15.06., 22.06.)
Laboratoria (1 grupa)	środy 17:05-19:40 (w okresie: 09.03.-08.06.)
Laboratoria (2 grupa)	czwartki 17:05-19:40 (w okresie: 10.03.-09.06.)

Wykłady przewidziane są dla Studentów Wydziału Informatyki i Telekomunikacji:

- I stopnia i II stopnia studiów
- **dla wszystkich kierunków studiów:** Cyberbezpieczeństwo, Informatyczne Systemy Automatyki, Informatyka Algorytmiczna, Informatyka Stosowana, Informatyka Techniczna, Inżynieria Systemów, Sztuczna Inteligencja, Teleinformatyka, Telekomunikacja, Zaufane Systemy Sztucznej Inteligencji

Wymagania wstępne w zakresie wiedzy, umiejętności i innych kompetencji: brak, na pierwszym wykładzie w dniu 2 marca 2022 r. przeprowadzony zostanie TEST WSTĘPNY z języka C++, na podstawie którego pracownicy firmy Nokia wybiorą osoby, które będą mogły wziąć udział w tegorocznym kursie (ograniczenia wynikają z dostępnej liczby miejsc w ramach dwóch grup zajęć laboratoryjnych)

Cele kursu

- Nabycie wiedzy w zakresie nowoczesnych metod rozwoju oprogramowania.
- Nabycie wiedzy w zakresie rozwoju oprogramowania przy użyciu języka C++.
- Nabycie wiedzy w zakresie prowadzenia projektów przy użyciu tzw. metodyk zwinnych.
- Zdobycie umiejętności korzystania z wybranych elementów języka C++ w praktyce.
- Wykształcenie dobrych nawyków programowania zorientowanego obiektowo.

Efekty kształcenia

Z zakresu wiedzy:

- zna założenia metodyk zwinnych w odniesieniu do projektów programistycznych (W01)
- zna wybrane wzorce projektowe stosowane w rozwoju oprogramowania (W02)
- zna podstawowe metody pracy z kodem obiektowo zorientowanym (W03)

Z zakresu umiejętności:

- potrafi posługiwać się zintegrowanym środowiskiem do rozwoju oprogramowania (U01)
- potrafi napisać testy jednostkowe przy użyciu wybranej platformy testowej (U02)
- potrafi korzystać z wybranych elementów języka C++ w praktyce (U03)

Z zakresu kompetencji społecznych:

- ma świadomość potrzeby dzielenia się wiedzą w ramach zespołu projektowego.
- rozumie potrzebę dbania o jakość oprogramowania w kontekście pracy zespołowej.
- rozumie konieczność stałego doskonalenia się w obliczu ciągłych zmian.

Treści programowe

Forma zajęć - wykład		Liczba godzin
Wy1	Jak realizować innowacyjne projekty na czas	3
Wy2	Metodyki zwinne w zmiennym środowisku projektowym	3
Wy3	Warsztat zwinnego programisty	3
Wy4	Budowanie złożonych systemów informatycznych	3
Wy5	Kolokwium	3
Suma godzin wykładów		15
Liczba godzin całkowitego nakładu pracy studenta		30
Liczba miejsc na wykładzie		48 osób lub więcej

Forma zajęć - laboratorium		Liczba godzin
La1	Środowisko programisty C++	3
La2	Zarządzanie pamięcią w C++	3
La3	Biblioteka STL w C++	3
La4	Nowoczesne C++	3
La5	Test Driven Development na przykładzie C++	3
La6	Tworzenie SOLIDnego kodu obiektowego w C++	3
La7	Wzorce projektowe na przykładzie C++	3
La8	Programowanie współbieżne w C++	3
La9	Optymalizacje w C++	3
La10	Coder Dojo w C++	3
Suma godzin laboratorium		30
Liczba godzin całkowitego nakładu pracy studenta		60
Liczba miejsc na laboratoriach		2 grupy po 24 osoby

Ocena osiągnięcia efektów kształcenia

Oceny	Numer efektu kształcenia	Sposób oceny osiągnięcia efektu kształcenia
F1 – w trakcie semestru	W01, W02, W03	aktywność na wykładach i laboratorium ocena z kolokwium (część teoretyczna)
F2 – w trakcie semestru	U01, U02, U03	aktywność na wykładach i laboratorium ocena z kolokwium (część praktyczna)
Ocena podsumowująca $P = 0.33 * F1 + 0.67 * F2$		
Uczestnicy kursy otrzymają imienne certyfikaty uczestnictwa zawierające oceny i treści programowe.		

Literatura

LITERATURA PODSTAWOWA:

- Stephen Prata – Język C++. Szkoła programowania.
- Scott Meyers – STL w praktyce. 50 sposobów efektywnego wykorzystania.

- Scott Meyers – Skuteczny, nowoczesny C++. 42 sposoby lepszego posługiwania się językami C++11 i C++14.
- Kent Beck – TDD. Sztuka tworzenia dobrego kodu.
- Robert C. Martin – Czysty kod. Podręcznik dobrego programisty.
- Zasoby sieci Internet w odniesieniu do przedstawianych tematów.

LITERATURA UZUPEŁNIAJĄCA:

- Bjarne Stroustrup – Język C++. Kompendium wiedzy.
- Nicolai M. Josuttis – C++. Biblioteka standardowa. Podręcznik programisty.
- Anthony Williams – Język C++ i przetwarzanie współbieżne w akcji
- Robert C. Martin – Zwinne wytwarzanie oprogramowania. Najlepsze zasady, wzorce i praktyki.
- Elisabeth Freeman, Eric Freeman, Bert Bates, Kathy Sierra – Wzorce projektowe. Rusz głową!
- Andrew Stellman – Agile. Przewodnik po zwinnych metodykach programowania.