

KARTA PRZEDMIOTU

Nazwa przedmiotu: Praktyczne zastosowania wzorców projektowych (InfAu-IDSI>SM1PZWZ19)

Nazwa w języku polskim:

Nazwa w jęz. angielskim: Practical Use of Software Design Patterns

Dane dotyczące przedmiotu:

Jednostka oferująca przedmiot: Wydział Automatyki, Elektroniki i Informatyki

Przedmiot dla jednostki: Politechnika Śląska

Domyślny typ protokołu dla przedmiotu:

ZAL

Język wykładowy:

polski

Strona WWW:

<https://platforma.polsl.pl/rau2/course/view.php?id=692>

Skrócony opis:

Celem przedmiotu jest nauczanie studentów, że napisanie aplikacji, która spełnia wymagania klienta to tylko jeden z elementów ich pracy. Równie ważne jest przygotowanie aplikacji w sposób zgodny z wzorcami projektowania, co ułatwia (lub wręcz umożliwia) jej testowanie i późniejsze wprowadzanie zmian. Ten aspekt jest często pomijany lub wprowadzany w sposób niewystarczający podczas studiów inżynierskich, gdzie główny nacisk kładziony jest na umiejętność stworzenia aplikacji a mniejszy na jej późniejsze pielęgnowanie. W ramach przedmiotu zaprezentowane zostaną zasady tworzenia oprogramowania, które spełnia ogólnie przyjęte kryteria jakościowe: zasady SOLID, zgodność ze wzorcami projektowymi. Celem przedmiotu jest wykazanie studentom, że „dobra” aplikacja to nie tylko aplikacja, która spełnia założenia funkcjonalne, ale przede wszystkim aplikacja łatwa w testowaniu i modyfikacji.

Opis:

ECTS: 2

Całkowita liczba godzin: 60h

Liczba godzin zajęć z bezpośrednim udziałem nauczycieli akademickich lub innych osób prowadzących zajęcia i studentów 35h w tym:

Wykład: 15h

Laboratorium: 15h

Konsultacje: 5h

Liczba godzin przeznaczonych na pracę własną studenta

Przygotowanie do zaliczenia wykładu: 10h

Przygotowanie do laboratoriów, przygotowanie sprawozdań: 15h

Wykład:

Podczas wykładu przedstawione są podstawowe problemy, z którymi spotyka się programista tworzący kod w języku obiektowym. Tematyka pierwszych wykładów związana jest z zasadami tworzenia oprogramowania SOLID (Single Responsibility, Open-Close, Liskov Substitution Principle, Interface Segregation, Dependency Inversion). Dla każdego problemu przedstawiony jest przykład i możliwe rozwiązania. Kolejne wykłady skupiają się na omówieniu wzorców projektowych opisanych w klasycznej pozycji Gamma et al. W szczególności dotyczy to: wzorców kreacyjnych (Factory Method, Factory, Abstract Factory, Prototype, Builder), wzorców strukturalnych (Decorator, Adapter, Strategy, Template Method, Proxy, Bridge), wzorców dotyczących kolekcji (Iterator, Composite, Flyweight), wzorców komunikacyjnych (Command, Observer, Mediator, Chain of Responsibility) oraz pozostałych wzorców behawioralnych (Memento, Interpreter, Visitor). Przykłady podczas zajęć przedstawiane są w języku Java, jednak poruszane zagadnienia dotyczą wszystkich języków obiektowych.

Laboratorium:

Podczas laboratorium studenci mają możliwość wykonywania zadań praktycznych polegających na refaktoringu kodu w sposób taki, aby był on zgodny z przedstawionymi podczas wykładu zasadami. Dużą wagę przykładą się do przekazania studentom nie tylko wiedzy teoretycznej, ale także do przekonania ich, że przedstawione zagadnienia mają bardzo duże znaczenie w praktyce.

Literatura:

Literatura podstawowa:

Erich Gamma, Richard Helm, Ralph Johnson, John M. Vlissides. Wzorce projektowe. Elementy oprogramowania obiektowego wielokrotnego użytku. Helion

Eric Freeman, Elisabeth Freeman, Kathy Sierra, Bert Bates. Head First Design Patterns. Edycja polska. Helion

Literatura uzupełniająca:

James William Cooper. Java. Wzorce projektowe. Helion

Efekty uczenia się:

Student zna wzorce projektowe GoF i rozumie ich znaczenie w procesie tworzenia oprogramowania (sprawozdanie z ćwiczenia laboratoryjnego, test zaliczeniowy) K2A_W07

Student zna zasady tworzenia oprogramowania SOLID (sprawozdanie z ćwiczenia laboratoryjnego, test zaliczeniowy) K2A_W07

Student potrafi zastosować odpowiednie wzorce projektowe do rozwiązania konkretnego problemu programistycznego (sprawozdanie z ćwiczenia laboratoryjnego, test zaliczeniowy) K2A_W14

Student posiada umiejętność oceny jakości kodu i jest w stanie zaproponować jego modyfikację poprawiającą jakość (sprawozdanie z ćwiczenia laboratoryjnego, test zaliczeniowy) K2A_U11, K2A_K01

Student rozumie podstawowe problemy tworzenia złożonych aplikacji (sprawozdanie z ćwiczenia laboratoryjnego, test zaliczeniowy) K2A_U12,

Student zna konsekwencje tworzenia kodu niskiej jakości (sprawozdanie z ćwiczenia laboratoryjnego, test zaliczeniowy) K2A_U12

Metody i kryteria oceniania:

Kurs składa się z wykładów i laboratoriów. Zgodnie z regulaminem studiów Politechniki Śląskiej wykłady nie są obowiązkowe, jednak studentów obowiązuje wiedza z wykładów. Podczas laboratoriów studenci realizują zadania polegające na przygotowaniu lub modyfikacji kodu w sposób zgodny z zasadami i wzorcami projektowymi. Efektem każdego laboratorium jest sprawozdanie. Sprawozdanie jest oceniane przez prowadzącego. Do zaliczenia przedmiotu wszystkie sprawozdania muszą być zaliczone na ocenę 3.0 lub wyższą.

Warunkiem zaliczenia przedmiotu jest:

- pozytywne oceny ze sprawozdań
- zaliczenie testu teoretycznego organizowanego na podczas ostatniego wykładu.

Ocena końcowa wyliczana jest jako średnia ocen z testu teoretycznego i średniej ocen z laboratoriów.

Sylabus obowiązuje od roku akademickiego 2024/25, a jego zawartość nie podlega zmianom w trakcie trwania semestru.

Praktyki zawodowe:**Punkty przedmiotu w cyklach:**

<bez przypisanego programu>			
Typ punktów	Liczba	Cykl pocz.	Cykl kon.
Europejski System Transferu Punktów (ECTS)	2	2020/2021-Z	