

# KARTA PRZEDMIOTU

## 1. Informacje ogólne

|   |                                    |
|---|------------------------------------|
| <b>Nazwa przedmiotu i kod (wg planu studiów):</b> | Systemy Rozproszone <b>D1_3</b>    |
| <b>Nazwa przedmiotu (j. ang.):</b>                | Distributing System                |
| <b>Kierunek studiów:</b>                          | Informatyka                        |
| <b>Specjalność/specjalizacja:</b>                 | Sieciowe systemy informatyczne     |
| <b>Poziom kształcenia:</b>                        | studia I stopnia                   |
| <b>Profil kształcenia:</b>                        | praktyczny (P)                     |
| <b>Forma studiów:</b>                             | studia stacjonarne, niestacjonarne |
| <b>Obszar kształcenia:</b>                        | nauki techniczne                   |
| <b>Dziedzina:</b>                                 | nauki techniczne                   |
| <b>Dyscyplina nauki:</b>                          | informatyka                        |
| <b>Koordinator przedmiotu:</b>                    | dr inż. Mariusz Świecicki          |

## 2. Ogólna charakterystyka przedmiotu

|   |   |
|---|---|
| <b>Przynależność do modułu:</b>                                 | kształcenia kierunkowego  |
| <b>Status przedmiotu:</b>                                       | obowiązkowy   |
| <b>Język wykładowy:</b>   | polski  |
| <b>Rok studiów, semestr:</b>                                    | III, 5,6  |
| <b>Forma i wymiar zajęć według planu studiów:</b>               | stacjonarne - wykład 30h, ćw. laboratoryjne 30h , ćw. projektowe 30h<br>niestacjonarne - wykład 15h, ćw. laboratoryjne 15h , ćw. projektowe 15h |
| <b>Interesariusze i instytucje partnerskie (nieobowiązkowe)</b> |   |
| <b>Wymagania wstępne / Przedmioty wprowadzające:</b>            | Programowanie niskopoziomowe; Umiejętność programowania w języku C/C++; Znajomość struktur danych   |

### 3. Bilans punktów ECTS

| Całkowita liczba punktów ECTS (wg planu studiów; 1 punkt =25-30 godzin pracy studenta, w tym praca na zajęciach i poza zajęciami):  | 6<br>(A+B)                         | stacjonarne | Niestacjonarne |
|---|------------------------------------|-------------|----------------|
|   |                                    |             |                |
| <b>A. Liczba godzin wymagających bezpośredniego udziału nauczyciela (kontaktowych, w czasie rzeczywistym, w tym testy, egzaminy etc) z podziałem na typy zajęć oraz całkowita liczba punktów ECTS osiąganych na tych zajęciach</b>  | wykład                             | 30          | 15             |
|   | laboratorium                       | 30          | 15             |
|   | projekt                            | 30          | 15             |
|   | egzamin                            | 2           | 2              |
|   | konsultacje                        | 3           | 10             |
| <b>W sumie:</b>   |                                    | 95          | 57             |
| <b>ECTS</b>   |                                    | 4,5         | 2,4            |
| <b>B. Poszczególne typy zadań do samokształcenia studenta (niewymagających bezpośredniego udziału nauczyciela) wraz z planowaną średnią liczbą godzin na każde i sumaryczną liczbą ECTS (np. praca w bibliotece, w sieci, na platformie e-learningowej, w laboratorium, praca nad projektem końcowym, przygotowanie ogólne; suma poszczególnych godzin powinna zgadzać się z liczbą ogólną)</b> | przygotowanie do kolokwium         | 10          | 10             |
|   | przygotowanie do laboratorium      | 5           | 5              |
|   | przygotowanie sprawozdań           | 5           | 5              |
|   | praca w sieci                      | 5           | 10             |
|   | praca na platformie e-learningowej | 0           | 10             |
|   | przygotowanie do egzaminu          | 10          | 10             |
|   | przygotowanie do konsultacji       | 2           | 5              |
|   | uzupełnienie/studiowanie notatek   | 2           | 5              |
|   | studiowanie zalecanej literatury   | 5           | 10             |
|   | <b>w sumie:</b>                    |             | 44             |
| <b>ECTS</b>   |                                    | 1,5         | 3,6            |
| <b>C. Liczba godzin praktycznych/laboratoryjnych w ramach przedmiotu oraz związana z tym liczba punktów ECTS (ta liczba nie musi być powiązana z liczbą godzin kontaktowych, niektóre zajęcia praktyczne/laboratoryjne mogą odbywać się bez udziału nauczyciela):</b>   | laboratorium                       | 15          | 15             |
|   | praca na platformie e-learningowej | 10          | 10             |
|   | przygotowanie do kolokwium         | 10          | 10             |
|   | egzamin                            | 2           | 2              |
|   | <b>ECTS</b>                        |             | 1,48           |

### 4. Opis przedmiotu

#### Cel przedmiotu:

Celem dydaktycznym nauczania powyższego przedmiotu jest zapoznanie studenta z problematyką zagadnień związanych ze z projektowaniem i realizacją systemów rozproszonych. W trakcie realizacji programu z tego przedmiotu student poznaje podstawowe pojęcia z zakresu systemów rozproszonych i programowania współbieżnego, następnie jest zapoznawany z budową systemów i środowisk programowych, które służą do implementacji tego typu systemów.

W trakcie ćwiczeń laboratoryjnych student poznaje technologie budowy i projektowania aplikacji rozproszonych, poczynając na pisaniu aplikacji wieloprocesowych i wielowątkowych, poprzez elementy synchronizacji, a kończąc na środowiskach programowych, które służą do realizacji aplikacji w środowisku obiektów rozproszonych.

**Metody dydaktyczne:** np. podające (wykład), problemowe (konwersatorium, seminarium), aktywizujące (symulacja, metoda przypadków itp. ), eksponujące (pokaz, film), praktyczne (ćwiczenia, metoda projektów itp) – pełniejszy wykaz poniżej (prosimy wybrać najstosowniejsze - jedną lub więcej, można dodać własne metody)

**Treści kształcenia** (w rozbiciu na formę zajęć (jeśli są różne formy) i najlepiej w punktach):

**Wykłady:**

1. Omówienie podstawowych równoległych architektur komputerowych w kontekście genezy systemów rozproszonych: systemy równoległe ze wspólną pamięcią – własności, architektura, systemy sieciowe – architektura, własności, systemy rozproszone – architektura, struktura, własności, Podstawowe własności systemu rozproszonego: dzielenie zasobu, skalowalność, przyspieszanie obliczeń, niezawodność, otwartość, przeźroczystość, otwartość systemu.
2. Podstawowe modele obliczeniowe stosowane przy realizacji aplikacji w środowisku rozproszonym. Zarządzanie zasobami w systemie rozproszonym. Zasoby w środowisku rozproszonym, mechanizmy współdzielenia. Zarządca zasobów- architektura klient – serwer, Zarządca zasobów- architektura obiektów rozproszonych,
3. Procesy i Wątki: Pojęcie procesu oraz wątku procesu, pojęcie wieloprocusowości i wielowątkowości w kontekście systemu operacyjnego i aplikacji, Właściwości wątku, właściwości procesu, Szeregowanie procesów, podstawowe algorytmy szeregowania i ich własności, Mechanizm tworzenie procesu w systemie operacyjnym UNIX (funkcja fork - język C
4. Synchronizacja procesów i wątków: pojęcie sekcji krytycznej, narzędzia umożliwiające synchronizację procesów i wątków: semafony, regiony, krytyczne. monitory, pamięć współdzielona. Problem producenta i konsumenta i jego realizacja przy użyciu semaforów.
5. Zakleszczenia: pojęcie zakleszczenia oraz warunki konieczne aby zjawisko zakleszczenia wystąpiło, Problem pięciu filozofów, Graf przydziału zasobów, Metody zapobiegania zakleszczeniom oraz metody wykrywania zakleszczeń,
6. Elementy komunikacji międzyprocesowej: Łącza komunikacyjne pipe, Fifo, kolejki komunikatorów, własności, proste przykłady – język C Semafony zmienne współdzielone w systemie POSIX - proste przykłady - język C, Architektura klient- serwer - własności, Architektura serwera iteracyjnego, Architektura serwera współbieżnego,
7. Systemy sieciowe – komunikacja gniazdowa: architektura systemów sieciowych – struktura i własności. Pojęcie protokołu sieciowego, model warstwowy protokołu sieciowego, Protokół OSI oraz protokół TCP/IP, Komunikacja strumieniowa oraz komunikacja datagramowa – własności, Pojęcie portu, pojęcie pary gniazdowej, Mechanizm ustanowienia połączenia – uzgadnianie trójfazowe, Mechanizm zakończenia połączenia: Architektura programu klienta i serwera funkcjonujących w oparciu protokół TCP i UDP, Budowa serwera iteracyjnego, Budowa serwera współbieżnego.
8. Zdalne wywoływanie Procedur (RPC) Problem reprezentacji danych w środowisku rozproszonym, Mechanizm translacji programów w środowisku w rozproszonym środowisku obliczeniowym, Mechanizm przetaczania danych zastosowany w systemie Sun RPC, Zdalne wywoływanie procedur na przykładzie systemu SUN RPC.
9. Środowisko obiektów rozproszonych: standard CORBA, model obiektu, język IDL, Omówienie mechanizmu obliczeń w środowisku CORBA na przykładzie prostej aplikacji klient-serwer, Serwisy CORBA, Budowa serwera BOA, Budowa serwera POA, Tryby implementacji serwera stosującego BOA
10. Metody i narzędzia synchronizacji w rozproszonym środowisku obliczeniowym. Problem synchronizacji czasu, metoda Christina, algorytm Berkley, Pojęcie czasu logicznego, Pojęcie czasu fizycznego, zegar logiczny, Algorytm porządkowania zdarzeń w przestrzeni czasu logicznego, Koordynacja rozproszona, sekcja krytyczna w środowisku rozproszonym, Metody implementacji sekcji krytycznej w środowisku rozproszonym – algorytm centralnego serwera, algorytm rozproszony z wykorzystaniem zegarów logicznych, pierścieniowy algorytm wzajem-

nego wykluczania, Pojęcie elekcji, algorytm Tyrana, pierścieniowy algorytm elekcji, Pojęcie zwielokrotnia, własności, architektury systemów stosujących zwielokrotnianie

### Ćwiczenia laboratoryjne:

1. Procesy i Wątki: Pojęcie procesu oraz wątku procesu, pojęcie wieloprocesowości i wielowątkowości w kontekście systemu operacyjnego i aplikacji, Właściwości wątku, właściwości procesu, Szeregowanie procesów, podstawowe algorytmy szeregowania i ich własności, Mechanizm tworzenie procesu w systemie operacyjnym *UNIX* (funkcja *fork* - język *C*) Budowa prostych aplikacji wielowątkowej w języku *C* w trakcie ćwiczeń laboratoryjnych.
2. Synchronizacja: Pojęcie sekcji krytycznej Narzędzia umożliwiające synchronizację procesów i wątków Semaforey, Regiony krytyczne, Monitory, Pamięć współdzielona. Problem *producenta i konsumenta*. : Pojęcie zakleszczenia oraz warunki konieczne aby zjawisko zakleszczenia wystąpiło, Problem *czytelników i pisarzy*, Problem *pięciu filozofów*, Graf przydziału zasobów, Metody zapobiegania zakleszczeniom oraz metody wykrywania zakleszczeń. Napisanie aplikacji rozwiązującej operującej na współdzielonych zasobach np. pliku w trakcie ćwiczeń laboratoryjnych
3. Komunikacja międzyprocesowa: Łączy komunikacyjne *pipe*, *Fifo*, kolejki komunikatów, własności, proste przykłady – język *C* Architektura klient- serwer - własności, Architektura serwera iteracyjnego, Architektura serwera współbieżnego. Realizacja w trakcie ćwiczeń laboratoryjnych prostych aplikacji klient-serwer wykorzystującej poszczególne mechanizmy komunikacji międzyprocesowej.
4. Systemy Sieciowe: Pojęcie protokołu sieciowego, model warstwowy protokołu sieciowego, Protokół TCP/IP, Komunikacja strumieniowa oraz komunikacja datagramowa – własności, Pojęcie portu, pojęcie pary gniazdowej, Mechanizm ustanowienia połączenia – uzgadnianie trójfazowe, Mechanizm zakończenia połączenia: Architektura programu klienta i serwera funkcjonujących w oparciu protokół TCP. Budowa serwera iteracyjnego, Budowa serwera współbieżnego opartego na: Procesach, Wątkach, Funkcje w Języku *C* : *socket*, *connect*, *accept*, *listen*, *bind*, *close*, *read*, *write*. Realizacja aplikacji klient serwera w języku *C* np. echa w trakcie ćwiczeń laboratoryjnych.
5. Własności komunikacji datagramowej, Budowa klienta pracującego w oparciu o protokół *udp*, Budowa serwera pracującego w oparciu protokół *udp*, Funkcje w Języku *C* : *socket*, *connect*, *listen*, *bind*, *close*, *read*, *write*, *recvfrom*, *sendto* w odniesieniu do protokołu *udp* Problemy powstające w komunikacji datagramowej. Tracenie datagramów – przyczyny i sposoby rozwiązania problemu, Brak sterowania przepływem - – przyczyny i sposoby rozwiązania problemu. Realizacje programów klienta i serwera w trakcie ćwiczeń laboratoryjnych wykorzystujących funkcje *sendto*, *recvfrom* ( w języku *C* )
6. Zdalne wywoływanie procedur *RPC*: Zasady budowy aplikacji klient – serwer korzystającej z mechanizmu *RPC*. Budowa odległego interfejsu, Implementacja odległego interfejsu, Budowa programu serwera dostarczającego zdalnych metod, Budowa programu klienta i mechanizm korzystania z zdalnego obiektu. Realizacja prostego programu klient – serwer korzystającego z mechanizmu *RPC*
7. Podstawowe techniki programowania w środowisku *CORBA*: Tryby implementacji serwera stosującego *BOA*. Wiele instancji na jednym serwerze, Posługiwanie się repozytorium: Obiekt o implementacji wspólnej, Obiekt o implementacji rozdzielczej, Trwałość danych, Migracja obiektów. Realizacja złożonego programu klient – serwer, gdzie serwer jest serwer wieloinstancyjnym

### Projekt:

**Realizacja aplikacji w wybranym środowisku programowym, która będzie wykorzystywała poniżej wymienione mechanizmy**

1. Komunikacja międzyprocesowa: Łączy komunikacyjne *pipe*, *Fifo*, kolejki komunikatów, wła-

sności, proste przykłady – język C Architektura klient- serwer - własności, Architektura serwera iteracyjnego, Architektura serwera współbieżnego. Realizacja w trakcie ćwiczeń laboratoryjnych prostych aplikacji klient-serwer wykorzystującej poszczególne mechanizmy komunikacji międzyprocesowej.

2. Systemy Sieciowe: Pojęcie protokołu sieciowego, model warstwowy protokołu sieciowego, Protokół TCP/IP, Komunikacja strumieniowa oraz komunikacja datagramowa – własności, Pojęcie portu, pojęcie pary gniazdowej, Mechanizm ustanowienia połączenia – uzgadnianie trójfazowe, Mechanizm zakończenia połączenia: Architektura programu klienta i serwera funkcjonujących w oparciu protokół TCP. Budowa serwera iteracyjnego, Budowa serwera współbieżnego opartego na: Procesach, Wątkach, Funkcje w Języku C : *socket, connect, accept, listen, bind, close, read, write*. Realizacja aplikacji klient serwera w języku C np. echa w trakcie ćwiczeń laboratoryjnych.
3. Własności komunikacji datagramowej, Budowa klienta pracującego w oparciu o protokół udp, Budowa serwera pracującego w oparciu protokół udp, Funkcje w Języku C : *socket, connect, listen, bind, close, read, write, recvfrom, sendto* w odniesieniu do protokołu *udp* Problemy powstające w komunikacji datagramowej. Tracenie datagramów – przyczyny i sposoby rozwiązania problemu, Brak sterowania przepływem - – przyczyny i sposoby rozwiązania problemu. Realizacje programów klienta i serwera w trakcie ćwiczeń laboratoryjnych wykorzystujących funkcje *sendto, recvfrom* ( w języku C )
4. Zdalne wywoływanie procedur RPC: Zasady budowy aplikacji klient – serwer korzystającej z mechanizmu RPC. Budowa odległego interfejsu, Implementacja odległego interfejsu, Budowa programu serwera dostarczającego zdalnych metod, Budowa programu klienta i mechanizm korzystania z zdalnego obiektu. Realizacja prostego programu klient – serwer korzystającego z mechanizmu RPC
5. Podstawowe techniki programowania w środowisku CORBA: Tryby implementacji serwera stosującego BOA. Wiele instancji na jednym serwerze, Posługiwanie się repozytorium: Obiekt o implementacji wspólnej, Obiekt o implementacji rozdzielczej, Trwałość danych, Migracja obiektów. Realizacja złożonego programu klient – serwer, gdzie serwer jest serwer wieloinstancyjnym

## 5. Efekty kształcenia i sposoby weryfikacji

**Efekty kształcenia** (w sumie wymienić ok. od 3 do 9 efektów - podać numery efektów z listy dla danego kierunku/specjalności – opublikowane na stronie uczelni; podać TYLKO te efekty (tam gdzie to możliwe i stosowne w trzech kategoriach, np. kompetencje społeczne mogą nie być realizowane w tym przedmiocie), na których osiągnięcie kładzie się nacisk w ramach przedmiotu, wybrane efekty kierunkowe powinny być bardziej szczegółowo sformułowane niż te dla całej specjalności, tak aby były weryfikowalne – dlatego mają osobne symbole jako efekty przedmiotu)

| Efekt przedmiotu<br>(kod przedmiotu + kod efektu kształcenia) | Student, który zaliczył przedmiot (spełnił minimum wymagań)  | Efekt kierunkowy                               |
|---|--|--|
| D1_3_W06,<br>D1_3_W07,<br>D1_3_W08,<br>D1_3_W14,<br>D1_3_W18, | <b>Wiedza:</b> <ol style="list-style-type: none"> <li>1. Zapoznanie studentów z budowa i struktura systemu zdecentralizowanego, z modelami obliczeniowymi które są wykorzystywane w systemach rozproszonych i sieciowych</li> <li>2. Zaznajomienie studentów z zasadami programowania współbieżnego i omówienie relewantnych zagadnień związanych z tą problematyką</li> <li>3. Implementacja problemów synchronizacji i komunikacji procesów z wykorzystaniem systemowych mechanizmów IPC</li> <li>4. Omówienie narzędzi – technologii służącej do budowy systemów</li> </ol> | K_W06,<br>K_W07,<br>K_W08,<br>K_W14,<br>K_W18, |

|  |  |  |
|--|--|--|
|  | zdecentralizowanych  |  |
| D1_3_U03,<br>D1_3_U14,<br>D1_3_U16,<br>D1_3_U17,<br>D1_3_U19,<br>D1_3_U24, | <b>Umiejętności</b> <ol style="list-style-type: none"> <li>Umiejętność posługiwania się językiem powłoki, przy rozhisteryzowaniu elementarnych problemów z zakresu administrowania systemem komputerowym</li> <li>Umiejętność rozwiązywania problemów z zakresu synchronizacji</li> <li>Umiejętność rozwiązywania problemów z zakresu komunikacji procesów z wykorzystaniem systemowych mechanizmów IPC</li> </ol> | K_U03,<br>K_U14,<br>K_U16,<br>K_U17,<br>K_U19,<br>K_U24, |
| D1_3_K01   | <b>Kompetencje społeczne</b> <ol style="list-style-type: none"> <li>Potrafi pracując w zespole zaprojektować i zaimplementować relacyjną bazę danych.</li> </ol>   | K_K04<br>K_K07   |

**Sposoby weryfikacji efektów kształcenia:**

*(np. dyskusja, gra dydaktyczna, zadanie e-learningowe, ćwiczenie laboratoryjne, projekt indywidualny/ grupowy, zajęcia terenowe, referat studenta, praca pisemna, kolokwium, test zaliczeniowy, egzamin, opinia eksperta zewnętrznego, etc. Dodać do każdego wybranego sposobu symbol zakładanego efektu, jeśli jest ich więcej)*

| Lp. | Efekt przedmiotu   | Sposób weryfikacji   | Ocena formująca – przykładowe sposoby jej wystawienia poniżej                   | Ocena końcowa przykładowe sposoby jej wystawienia poniżej |
|-----|--|--|---|---|
| 1   | D1_3_W06,<br>D1_3_W07,<br>D1_3_W08,<br>D1_3_W14,<br>D1_3_W18,              | Egzamin: - sprawdzian wiedzy, sprawdzian umiejętności – test zawierający pytania otwarte i zamknięte.  | Ocena z egzaminu – z części testowej oraz z części zawierającej pytania otwarte | Średnia ocen formujących                                  |
| 2   | D1_3_U03,<br>D1_3_U14,<br>D1_3_U16,<br>D1_3_U17,<br>D1_3_U19,<br>D1_3_U24, | ćwiczenia laboratoryjne – sprawdziany weryfikacja umiejętności praktyczne i teoretyczne po zakończeniu każdej grupy zagadnień za praktyczne studenta | Ocena z sprawdzianu   | Średnia ocen formujących                                  |
| 3   | D1_3_K01   | Dyskusja na temat przyjętych rozwiązań dla zadanego problemu z zakresu systemów rozproszonych  | Oceny za aktywność  | Średnia ocen formujących                                  |

**Kryteria oceny (oceny ,3.0 powinny być równoważne z efektami kształcenia, choć mogą być bardziej szczegółowo opisane):**

| w zakresie wiedzy |   | Efekt kształcenia   |
|-------------------|---|---|
| Na ocenę 3,0      | Student uzyskuje 50-59% maksymalnej liczby punktów z części dotyczących efektów wydzielonej z kolokwium i egzaminu końcowego      | D1_3_W06,<br>D1_3_W07,<br>D1_3_W08,<br>D1_3_W14,<br>D1_3_W18, |
| Na ocenę 5,0      | Student uzyskuje powyżej 89% maksymalnej liczby punktów z części dotyczących efektów wydzielonej z kolokwium i egzaminu końcowego | D1_3_W06,<br>D1_3_W07,<br>D1_3_W08,<br>D1_3_W14,<br>D1_3_W18, |

| <b>w zakresie umiejętności</b>  |  |  |
|---|--|--|
| Na ocenę 3,0  | Student zalicza wszystkie ćwiczenia laboratoryjne związane z efektami kształcenia, oddaje projekty ze wszystkich zadań i zalicza kolokwia (otrzymuje średnia z kolokwiów 3.0-3.25)                           | D1_3_U03,<br>D1_3_U14,<br>D1_3_U16,<br>D1_3_U17,<br>D1_3_U19,<br>D1_3_U24, |
| Na ocenę 5,0  | Student zalicza wszystkie ćwiczenia laboratoryjne związane z efektami kształcenia, oddaje projekty ze wszystkich zadań i zalicza kolokwia (otrzymuje średnia z kolokwiów powyżej 4.75)                       | D1_3_U03,<br>D1_3_U14,<br>D1_3_U16,<br>D1_3_U17,<br>D1_3_U19,<br>D1_3_U24, |
| <b>w zakresie kompetencji społecznych</b>   |  |  |
| Na ocenę 3,0  | Potrafi pracując w zespole zaprojektować prosty system rozproszony o architekturze sieciowej   | D1_3_K01   |
| Na ocenę 5,0  | Potrafi pracując w zespole zaprojektować i zrealizować system zdecentralizowany dla wybranego zagadnienia, który będzie wyposażony w elementy równoważenia obciążenia i elementy współbieżnego przetwarzania | D1_3_K01   |
| <b>Kryteria oceny końcowej</b> średniej arytmetycznej ocen uzyskanych z egzaminu i laboratorium   |  |  |
| <b>Zalecana literatura</b> (w podziale na literaturę podstawową i uzupełniającą):<br><b>Podstawowa:</b>   |  |  |
| <ol style="list-style-type: none"> <li>1. Walery Rogoza , Metody i środki projektowania obiektów interoperabilnych , ISBN: 83-60434-07-7</li> <li>2. <i>Coulouris G, Dollimore J. Kindberg T:</i> Systemy rozproszone – podstawy I projektowanie, WNT</li> <li>3. <i>Stevens W.R:</i> Unix programowanie usług sieciowych Api: gniazda i XTI; WNT</li> <li>4. M.Mitchell, J. Oldham, A.Samuel — LINUX Programowanie dla zaawansowanych, Warszawa, 2002, Wydawnictwo RM</li> <li>5. <i>Stevens W.R:</i> Unix programowanie usług sieciowych komunikacja międzyprocesowa, WNT</li> <li>6. <i>Sawerwain M.:</i> CORBA programowanie w praktyce, MIKOM</li> </ol> |  |  |

#### **Informacje dodatkowe:**

|  |
|--|
| <p><b>Dodatkowe obowiązki prowadzącego wraz z szacowaną całkowitą liczbą godzin:</b></p> <p>Konsultacje – 10 godzin</p> <p>Przygotowanie stanowisk laboratoryjnych – 15 godzin</p> <p>Przygotowanie ćwiczeń e-learningowych - 10 godzin</p> <p>Przygotowanie i poprawa egzaminu – 10 godzin</p> <p><b>W sumie: 45 godzin</b></p> |
|--|