

OPIS PRZEDMIOTU/MODUŁU KSZTAŁCENIA (SYLABUS)

1.	Nazwa przedmiotu/modułu w języku polskim Praktyczne aspekty nowoczesnej inżynierii oprogramowania	
2.	Nazwa przedmiotu/modułu w języku angielskim Practical aspects of modern software engineering	
3.	Jednostka prowadząca przedmiot Instytut Fizyki Teoretycznej	
4.	Kod przedmiotu/modułu	
5.	Rodzaj przedmiotu/modułu (<i>obowiązkowy lub fakultatywny</i>) Fakultatywny	
6.	Kierunek studiów Fizyka, Fizyka techniczna	
7.	Poziom studiów (<i>I lub II stopień lub jednolite studia magisterskie</i>) I	
8.	Rok studiów (<i>jeśli obowiązuje</i>)	
9.	Semestr (<i>zimowy lub letni</i>) Letni	
10.	Forma zajęć i liczba godzin Laboratorium komputerowe (15 x 2h)	
11.	Imię, nazwisko, tytuł/stopień naukowy osoby prowadzącej zajęcia Mateusz Wroński, dr inż., Mateusz Jaworski, mgr inż.	
12.	Wymagania wstępne w zakresie wiedzy, umiejętności i kompetencji społecznych dla przedmiotu/modułu oraz zrealizowanych przedmiotów Wskazana znajomość programowania obiektowego w Java na poziomie podstawowym.	
13.	Cele przedmiotu Celem przedmiotu jest zaznajomienie studentów z aktualnie stosowanymi metodami pracy w grupie inżynierów oprogramowania oraz praktyczne zastosowanie zdobytej wiedzy w pracy grupy studentów nad wspólnym, prostym projektem programistycznym. Dodatkowym, pobocznym celem przedmiotu jest poszerzenie wiedzy studentów o języku programowania Java.	
14.	Zakładane efekty kształcenia Znajomość nowoczesnych metod prowadzenia projektów programistycznych. Praktyczna znajomość zasad działania podstawowych metod efektywnego tworzenia i refaktoryzacji kodu oprogramowania, użytkowania systemów kontroli	Symbole kierunkowych efektów kształcenia: K_W09, K_U14 (fizyka) K_W18, K_U19 (fizyka techniczna)

	wersji oraz ciągłej integracji.																									
15.	<p>Treści programowe</p> <ul style="list-style-type: none"> i. Klasyczne i współczesne metody prowadzenia projektów programistycznych <ul style="list-style-type: none"> a. Modele kaskadowe b. KanBan, Scrum - programowanie zwinne ii. Techniki pisania czystego kodu oprogramowania <ul style="list-style-type: none"> a. Utrzymanie kodu i refaktoring b. Statyczna analiza kodu iii. Automatyczne testy kodu i aplikacji – testy jednostkowe i funkcjonalne iv. Rozproszone i scentralizowane systemy kontroli wersji <ul style="list-style-type: none"> a. Git b. Subversion c. Konflikty kodu i ich rozwiązywanie v. Automatyzacja procesu budowy oprogramowania vi. Ciągła integracja w projekcie informatycznym 																									
16.	<p>Zalecana literatura (<i>podręczniki</i>)</p> <p>K. Schwaber, J. Sutherland: The Scrum Guide: Przewodnik po scrumie: Reguły gry</p>																									
17.	<p>Forma zaliczenia poszczególnych komponentów przedmiotu/modułu, sposób sprawdzenia osiągnięcia zamierzonych efektów kształcenia:</p> <p>wykład: seminarium: laboratorium: Grupowy projekt programistyczny rozwijany w trakcie semestru konwersatorium: inne:</p>																									
18.	<p>Język wykładowy</p> <p>Polski</p>																									
19.	<p>Obciążenie pracą studenta</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: center;">Forma aktywności studenta</th> <th style="text-align: center;">Średnia liczba godzin na zrealizowanie aktywności</th> </tr> </thead> <tbody> <tr> <td>Godziny zajęć (wg planu studiów) z nauczycielem:</td> <td></td> </tr> <tr> <td>- wykład:</td> <td style="text-align: right;">-</td> </tr> <tr> <td>- ćwiczenia:</td> <td style="text-align: right;">-</td> </tr> <tr> <td>- laboratorium:</td> <td style="text-align: right;">30</td> </tr> <tr> <td>- inne:</td> <td style="text-align: right;">-</td> </tr> <tr> <td>Praca własna studenta:</td> <td></td> </tr> <tr> <td>- przygotowanie do zajęć laboratoryjnych:</td> <td style="text-align: right;">30</td> </tr> <tr> <td>- czytanie wskazanej literatury:</td> <td style="text-align: right;">15</td> </tr> <tr> <td>- przygotowanie projektu zaliczeniowego:</td> <td style="text-align: right;">-</td> </tr> <tr> <td>Suma godzin</td> <td style="text-align: right;">75</td> </tr> <tr> <td>Liczba punktów ECTS</td> <td style="text-align: right;">3</td> </tr> </tbody> </table>		Forma aktywności studenta	Średnia liczba godzin na zrealizowanie aktywności	Godziny zajęć (wg planu studiów) z nauczycielem:		- wykład:	-	- ćwiczenia:	-	- laboratorium:	30	- inne:	-	Praca własna studenta:		- przygotowanie do zajęć laboratoryjnych:	30	- czytanie wskazanej literatury:	15	- przygotowanie projektu zaliczeniowego:	-	Suma godzin	75	Liczba punktów ECTS	3
Forma aktywności studenta	Średnia liczba godzin na zrealizowanie aktywności																									
Godziny zajęć (wg planu studiów) z nauczycielem:																										
- wykład:	-																									
- ćwiczenia:	-																									
- laboratorium:	30																									
- inne:	-																									
Praca własna studenta:																										
- przygotowanie do zajęć laboratoryjnych:	30																									
- czytanie wskazanej literatury:	15																									
- przygotowanie projektu zaliczeniowego:	-																									
Suma godzin	75																									
Liczba punktów ECTS	3																									

***objaśnienie symboli:**

K (przed podkreśleniem) - kierunkowe efekty kształcenia
W - kategoria wiedzy
U - kategoria umiejętności
K (po podkreśleniu) - kategoria kompetencji społecznych
01, 02, 03 i kolejne - numer efektu kształcenia

COURSE/MODULE DESCRIPTION (SYLLABUS)

1.	Course/module Practical aspects of a modern software engineering	
2.	University department Institute of Theoretical Physics	
3.	Course/module code	
4.	Course/module type – mandatory (compulsory) or elective (optional) Elective	
5.	University subject (programme/major) physics, technical physics	
6.	Degree: (<i>master, bachelor</i>) Bachelor	
7.	Year	
8.	Semester (<i>autumn, spring</i>) Spring	
9.	Form of tuition and number of hours computer lab (15 x 2h)	
10.	Name, Surname, academic title Mateusz Wroński, dr inż., Mateusz Jaworski, mgr inż.	
11.	Initial requirements (knowledge, skills, social competences) regarding the course/module and its completion Basic knowledge on object programming in Java.	
12.	Objectives The aim of the course is acquiring the knowledge on the latest techniques of work in a group of software engineers and it's practical usage by working in a group of students on a small and simple software project. Additional advantage is extending the knowledge on programming in Java.	
13.	Learning outcomes Knowledge on basic methods of software projects development and management. Skills of usage of tools for versioning control, continuous integration, basic methods of writing effective, clean code and refactorization.	Outcome symbols: K_W09, K_U14 (physics) K_W18, K_U19 (technical physics)
14.	Content I. Modern methods of software projects development and management <ul style="list-style-type: none"> • software development methodologies • KanBan, Scrum and Agile philosophy II. Techniques of writing clean and effective code <ul style="list-style-type: none"> • code maintenance and refactoring • statical code analysis III. Automated code testing – unit and functional IV. Distributed and centralized version control systems	

	<ul style="list-style-type: none"> • Git • Subversion • Code conflicts and merging <p>III. Automated code compilation and continuous integration IV. Parallel and distributed programming in java</p>																								
15.	<p>Recommended literature</p> <p>K. Schwaber, J. Sutherland: The Scrum Guide: Scrum.org and Scrum</p>																								
16.	<p>Ways of earning credits for the completion of a course /particular component, methods of assessing academic progress:</p> <p>lecture: class: laboratory: Group software project run for whole semester seminar: other:</p>																								
17.	<p>Language of instruction</p> <p>Polish</p>																								
18.	<table border="1"> <thead> <tr> <th>Activity</th> <th>Average number of hours for the activity</th> </tr> </thead> <tbody> <tr> <td>Hours of instruction (as stipulated in study programme) :</td> <td></td> </tr> <tr> <td>- lecture:</td> <td>-</td> </tr> <tr> <td>- classes:</td> <td>-</td> </tr> <tr> <td>- laboratory:</td> <td>30</td> </tr> <tr> <td>- other:</td> <td>-</td> </tr> <tr> <td>student's own work, e.g.:</td> <td></td> </tr> <tr> <td>- preparation before labs</td> <td>30</td> </tr> <tr> <td>- reading set literature:</td> <td>15</td> </tr> <tr> <td>- preparing passing project:</td> <td></td> </tr> <tr> <td>Hours</td> <td>75</td> </tr> <tr> <td>Number of ECTS</td> <td>3</td> </tr> </tbody> </table>	Activity	Average number of hours for the activity	Hours of instruction (as stipulated in study programme) :		- lecture:	-	- classes:	-	- laboratory:	30	- other:	-	student's own work, e.g.:		- preparation before labs	30	- reading set literature:	15	- preparing passing project:		Hours	75	Number of ECTS	3
Activity	Average number of hours for the activity																								
Hours of instruction (as stipulated in study programme) :																									
- lecture:	-																								
- classes:	-																								
- laboratory:	30																								
- other:	-																								
student's own work, e.g.:																									
- preparation before labs	30																								
- reading set literature:	15																								
- preparing passing project:																									
Hours	75																								
Number of ECTS	3																								

* Key to symbols:

K (before underscore) - learning outcomes for the programme

W - knowledge

U - skills

K (after underscore) - social competences

01, 02, 03 and subsequent - consecutive number of learning outcome