#### card of course

|  |  |
| --- | --- |
| Subject name | Software Engineering |

1. The placement of the subject in the study system

|  |  |
| --- | --- |
| 1.1. Field of study | Computer science |
| 1.2. Form and path of study | Full-time/Part-time |
| 1.3. Level of education | First-cycle studies |
| 1.4. Study profile | Practical |

|  |  |
| --- | --- |
| 1. 5. Specialty | - |
| 1.6. Subject Coordinator | Dr inż. Kamil Żyła |

2. General characteristics of the subject

|  |  |
| --- | --- |
| 2.1. Belonging to a subject group | Directional/Practical |
| 2.2. Number of ECTS | 3 |
| 2.3. Language of lectures | English |
| 2.4. Semesters in which the subject is taught | IV |
| 2.5.Criteria for selecting course participants | - |

1. Learning outcomes and course delivery
   1. Subject Objectives

|  |  |
| --- | --- |
| No. | Subject Objectives |
|
| C1 | Providing students with knowledge of software engineering, in the field of organizing the course of a programming project, object-oriented software design, and supporting IT tools. |
| C2 | Providing students with knowledge and skills in using the UML language. |
| C3 | To provide students with basic theoretical and practical knowledge of design patterns. |

* 1. Subject-specific learning outcomes, divided into knowledge , skills and competences , with reference to the directional learning outcomes

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Description of subject  learning outcomes | Reference to  directional learning outcomes (symbols) | Method of implementation (mark "X") | | | |
| ST | | NST | |
| Classes at the University | Activities on  the platform | Classes at the University | Activities on  the platform |
| After passing the course, the student knows and understands **the knowledge** | | | | | | |
| W1 | the importance and role of typical UML diagrams, as well as UML notation elements, in the software development process | INF\_W07  INF\_W20 INF\_W21 |  | X |  | X |
| W2 | possibilities and methods of using IT tools supporting software design |  | X |  | X |
| W3 | how individual functional and non-functional requirements are translated into UML diagrams |  | X |  | X |
| W4 | what are the components of a software project |  | X |  | X |
| W5 | the idea and basic division of design patterns |  | X |  | X |
| After passing the course, the student is **able** to: | | | | | | |
| U1 | use UML syntax to draw diagrams | INF\_U06 INF\_U10 INF\_U15  INF\_U20 INF\_U21 | X |  | X |  |
| U2 | use tools to support drawing diagrams in UML | X |  | X |  |
| U3 | prepare a description of the "real world" as well as a specification of the requirements of the designed software | X |  | X |  |
| U4 | create and understand software documentation oriented towards the use of UML | X |  | X |  |
| After completing the course, the student is ready to take part in **social competences.** | | | | | | |
| K1 | critically assess the quality of your work and seek alternative solutions | INF\_K03 INF\_K06 | X |  | X |  |
| K2 | uses software engineering knowledge in social and technical discourse | X |  | X |  |

3.3. Forms of teaching and their number of hours - Full-time studies (ST), Part-time studies (NST)

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Path | Lecture | Exercises | Design | Workshop | Laboratory | Seminar | Lecturer | Classes conducted using distance learning methods and techniques in the form of a lecture | Other | **ECTS points** |
| **ST** |  |  | 30 |  |  |  |  | 10 |  | 3 |
| **NST** |  |  | 15 |  |  |  |  | 10 |  | 3 |

3.4. Content of education (separately for each form of classes: (W, ĆW, PROJ, WAR, LAB, LEK, OTHER). It should be marked (X) how the given content will be implemented (classes at the university or classes on the e-learning platform conducted using distance learning methods and techniques)

TYPE OF CLASS: LECTURE

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Content of the course | Reference to subject-specific learning outcomes | Method of implementation (mark "X") | | | |
| ST | | NST | |
| **Classes at the University** | **Activities on  the platform** | **Classes at the University** | **Activities on  the platform** |
| 1. | Introduction to the subject - program content, basic concepts, importance and place of the UML language in the software development process, supporting tools. | W1, W2, W4 |  | X |  | X |
| 2. | Software requirements specification in the context of using the UML language. Components of a software project using the UML language. | W3, W4 |  | X |  | X |
| 3. | UML package diagram. | W1, W3 |  | X |  | X |
| 4. | Use case diagram in UML. Use case execution scenarios. | W1, W3 |  | X |  | X |
| 5. | Class diagram in UML. | W1, W3 |  | X |  | X |
| 6. | Sequence diagram in UML. | W1, W3 |  | X |  | X |
| 7. | Activity diagram in UML. | W1, W3 |  | X |  | X |
| 8. | State diagram in UML. | W1, W3 |  | X |  | X |
| 9. | Selected design patterns. | W2, W5 |  | X |  | X |
| 10. | Implementation diagram in UML. | W1, W3 |  | X |  | X |
| 11. | Summary of classes and discussion of grades. |  |  | X |  | X |

TYPE OF CLASS: PROJECT

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| No. | Content of the course | Reference to subject-specific learning outcomes | Method of implementation (mark "X") | | | |
| ST | | NST | |
| **Classes at the University** | **Activities on  the platform** | **Classes at the University** | **Activities on  the platform** |
| 1. | Introduction to the subject - program content, basic concepts, specifics of work, supporting tools, components of a software project. | U2, U4, K1, K2 | X |  | X |  |
| 2. | Description of the "real world" and architecture of the designed software. Specification of requirements and use cases for the software. | U3, U4, K1, K2 | X |  | X |  |
| 3. | Building use case diagrams in UML. Use case implementation scenarios. | U1, U2, U4, K1 | X |  | X |  |
| 4. | Building class diagrams in UML. | U1, U2, U4, K1 | X |  | X |  |
| 5. | Building sequence diagrams in UML. | U1, U2, U4, K1 | X |  | X |  |
| 6. | Building activity diagrams in UML. | U1, U2, U4, K1 | X |  | X |  |
| 7. | Building state diagrams in UML. | U1, U2, U4, K1 | X |  | X |  |
| 8. | Selected design patterns. | U1, U2, U4, K1 | X |  | X |  |
| 9. | Summary of classes and discussion of grades. |  | X |  | X |  |

3.5. Methods of verifying learning outcomes (indication and description of methods of conducting classes and verification of achievement of learning outcomes and method of documentation)

|  |  |  |  |
| --- | --- | --- | --- |
| Subject Effects | Teaching methods | Methods of verifying learning outcomes | Documentation methods |
| KNOWLEDGE | | | |
| W1-W5 | lecture, discussion | Passing the lectures: exam on the topics covered in the lecture | Graded exam sheet |
| SKILLS | | | |
| U1-U4 | Discussion, preparation of projects | Project credit: software engineering project concerning program content listed in the syllabus. Detailed guidelines regarding the scope of work provided to students during classes by the instructor | Rated project |
| SOCIAL COMPETENCES | | | |
| K1-K2 | Discussion, preparation of projects | Project credit: software engineering project concerning program content listed in the syllabus. Detailed guidelines regarding the scope of work provided to students during classes by the instructor | Rated project |

3.6. Assessment criteria for the achieved learning outcomes

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Learning effect | For a grade of 3 or "pass."  the student knows and understands/is able to/is ready to | For a grade of 3.5, the student knows and understands/is able to/is ready to | For a grade of 4, the student knows and understands/is able to/is ready to | For a grade of 4.5, the student knows and understands/is able to/is ready to | For a grade of 5, the student knows and understands/is able to/is ready to |
| W | 51-60% of knowledge indicated in learning outcomes | 61-70% of knowledge indicated in learning outcomes | 71-80% of knowledge indicated in learning outcomes | 81-90% of knowledge indicated in learning outcomes | 91-100% of knowledge indicated in learning outcomes |
| U | 51-60% of skills indicated in learning outcomes | 61-70% of skills indicated in learning outcomes | 71-80% of skills indicated in learning outcomes | 81-90% of skills indicated in learning outcomes | 91-100% of skills indicated in learning outcomes |
| K | 51-60% of skills indicated in learning outcomes | 61-70% of skills indicated in learning outcomes | 71-80% of skills indicated in learning outcomes | 81-90% of skills indicated in learning outcomes | 91-100% of skills indicated in learning outcomes |

3.7. Literature

**Basic**

1. Vliet Hans van, Software engineering: Principles and practice, John Wiley, Chichester, 2008
2. Michał Śmiałek, Kamil Rybiński, "Inżynieria oprogramowania w praktyce. Od wymagań do kodu z językiem UML", Helion, 2023
3. Eric Freeman, Elisabeth Robson, "Wzorce projektowe. Rusz głową! Tworzenie rozszerzalnego i łatwego w utrzymaniu oprogramowania obiektowego. Wydanie II", Helion, 2021

**Supplementary**

1. Techniczna dokumentacja języka UML: <https://www.omg.org/spec/UML/2.5.1/About-UML>
2. Wzorce projektowe w języku Java: <https://www.tutorialspoint.com/design_pattern>

4. Student workload - ECTS points balance

|  |  |  |
| --- | --- | --- |
| **Types of student activity** | **Student Load** | |
| **ST** | **NST** |
| **Classes requiring direct contact between the student and the academic teacher at the university premises** | **40** | **25** |
| Classes included in the study plan | 40 | 25 |
| **Student's own work** | **35** | **50** |
| Ongoing preparation for classes, preparation of project work/presentations/etc. | 15 | 25 |
| Preparation for passing classes | 20 | 25 |
| **TOTAL STUDENT HOURLY LOAD** | **75** | **75** |
| **Number of ECTS points** | **3** | **3** |

|  |  |
| --- | --- |
| Last change date | 30/09/2024 |
| The changes were introduced | INF Education Quality Team |
| The changes were approved | Arkadiusz Gwarda, M.A. |