#### card of course

|  |  |
| --- | --- |
| Subject name | Programming in JAVA |

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 | Web Technologies and the Internet of Things |
| 1.6. Subject Coordinator | Dr Barbara Gocłowska |

2. General characteristics of the subject

|  |  |
| --- | --- |
| 2.1. Belonging to a subject group | Optional/practical |
| 2.2. Number of ECTS | 5 |
| 2.3. Language of lectures | English |
| 2.4. Semesters in which the subject is taught | III |
| 2.5.Criteria for selecting course participants | For the specialization: Web Technologies and Internet of Things |

1. Learning outcomes and course delivery
	1. Subject Objectives

|  |  |
| --- | --- |
| No. | Subject Objectives |
|
| C1 | Acquiring knowledge of object-oriented programming |
| C2 | Gaining Java programming skills |
| C3 | Acquiring skills in creating window applications |

* 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 effectslearning (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 | Fundamentals of object-oriented programmingwego (class, object, class body, me-toda) | INF\_W08INF\_W20 | X |  |  | X |
| W2 | Creating classes, interfaces ( classes)wilding, interface implementation) | X |  |  | X |
| W3 | Implementation of methods | X |  |  | X |
| W4 | Rules for using polymorphism | X |  |  | X |
| After passing the course, the student is **able** to: |
| U1 | Compile and run single-classowls program | INF\_U15 INF\_U19 INF\_U20 INF\_U23 | X |  | X |  |
| U2 | Create simple programs , alsowindow (Swing) | X |  | X |  |
| U3 | Implement a two-tier applicationstate (database) | X |  | X |  |
| U4 | Design applications usingthe right tools | X |  | X |  |
| After completing the course, the student is ready to take part in **social competences.** |
| K1 | Collaborate in a team working on an application with division into roles | INF\_K04 | X |  | X |  |
| K2 | Becoming aware of the barriers that hinder interpersonal communication | X |  | X |  |
| K3 | Seek help in solving problems – on the Internet and in Open AI (preference for Java language is Google Bart) | 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** | 20 |  |  |  | 30 |  |  |  |  | 5 |
| **NST** |  |  |  |  | 20 |  |  | 10 |  | 5 |

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. | Java as a universal programming language. Java cross-platform.Types of Java programs .Basic structure of a Java program. Packages. | **W1** | **X** |  |  | **X** |
| 2. | Data types. Basic language operators and control statements. | **W1** | **X** |  |  | **X** |
| 3. | Programming environment. Compilation and execution. Object-oriented programming in Java. Abstract methods and classes, interfaces.Error handling : exception handling scheme , exception classes . | **W1** | **X** |  |  | **X** |
| 4. | Graphical user interface. AWT and Swing components: class hierarchy, component properties . Containers and their decompositions. Event handling model: event sources , listeners, actions. | **W2** | **X** |  |  | **X** |
| 5. | Handling data streams: the concept of a stream, the class hierarchy of stream objects . | **W3, W2** | **X** |  |  | **X** |
| 6. | Access to data via the JDBC interface. | **W4** | **X** |  |  | **X** |
| 7. | Summary of classes and discussion of grades. |  | X |  |  | X |

TYPE OF CLASS: LABORATORY

|  |  |  |  |
| --- | --- | --- | --- |
| 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. | Creating objects | **U1, U2** | **X** |  | **X** |  |
| 2. | Manipulating variables in sample exercises | **U1** | **X** |  | **X** |  |
| 3. | Implementation of conditional statements | **U1, U2** | **X** |  | **X** |  |
| 4. | How to Create an App with Instructions | **U2** | **X** |  | **X** |  |
| 5. | Inheritance and polymorphism - tasks | **U1, U2, U3 U4** | **X** |  | **X** |  |
| 6. | Abstract classes - examples | **U4** | **X** |  | **X** |  |
| 7. | Windowed Applications - Creating Applications | **U1, U2, U3** | **X** |  | **X** |  |
| 8. | System Managers - Their Implementation | **U3** | **X** |  | **X** |  |
| 9. | Listening to an event. Reacting to an event - tasks to solve | **U1, U2, U3 U4** | **X** |  | **X** |  |
| 10. | Collections - use in examples | **U2** | **X** |  | **X** |  |
| 11. | JDBC - implementation | **U4, K1, K2** | **X** |  | **X** |  |
| 12. | ResultSet Interface - API - Effective Use in Examples | **U4, K1, K2** | **X** |  | **X** |  |
| 13. | Enumeration classes - implementation | **U4** | **X** |  | **X** |  |
| 14. | Summary of classes and discussion of grades - credit application |  | 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)

Verification methods:

Lecture credit - oral credit testing the knowledge necessary to solve problems that arise during the creation of a credit program (defense of the prepared application according to the guidelines provided by the instructor). The minimum is an extensive desktop application, the next level is a window application of varying complexity. In the case of an extensive application, data is stored in a file or, better yet, in any database.

Laboratory credit: Two tests To pass the subject, the student must obtain at least 51% of the sum of points from both tests. The total number of points from both tests will be calculated based on the results achieved, with the points from each test being weighted proportionally to the number of points possible to obtain in that test.

|  |  |  |  |
| --- | --- | --- | --- |
| Subject Effects | Teaching methods | Methods of verifying learning outcomes | Documentation methods |
| KNOWLEDGE |
| W1-W4 | Lecture using the giving method , problems and their solution live (code and its execution). Methods of seeking help in solving problems . | Passing the lectures - oral examination - defense of the prepared application according to the guidelines given by the lecturer. | Oral exam transcript and application archived on the platform |
| SKILLS |
| U1-U4 | Doing exercises | Two colloquia (described above) | Archived Colloquia |
| SOCIAL COMPETENCES |
| K1-K3 | Doing exercises | Two colloquia (described above) | Archived Colloquia |

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. Kathy Sierra, Bert Bates Java Rusz głową - dostępne w Internecie w wersji angielskiej i polskiej
2. Herbert Schildt Java. Kompedium wiedzy, wyd IX Helion

**Supplementary**

1. Documentation: <https://docs.oracle.com/javase/tutorial/>
2. Documentation: <https://www.udemy.com/course/java-tutorial/>

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** | **50** | **30** |
| Classes included in the study plan | 50 | 30 |
| **Student's own work** | **75** | **95** |
| Ongoing preparation for classes, preparation of project work/presentations/etc. | 35 | 45 |
| Preparation for passing classes | 40 | 50 |
| **TOTAL STUDENT HOURLY LOAD** | **125** | **125** |
| **Number of ECTS points** | **5** | **5** |

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