The syllabus of the discipline

|  |  |  |
| --- | --- | --- |
| № | Field name | Detailed content, comments |
|  | Name of the faculty | Faculty of Computer Science |
|  | The level of higher education | Bachelor's |
|  | Code and title of specialty | 126 Information systems and technologies |
|  | The type and title of the educational program | EPP «Information technologies of the Internet of Things» |
|  | Code and title of the discipline | \_\_\_OOP\_\_\_ Object-oriented programming |
|  | Number of ECTS credits | 7 |
|  | The structure of the course (distribution by type and hours of training) | 1st semester: 24 h. – 12 lectures, 24 h.– 6 practical works, 8 h. – 4 consultations, 64 h. – independent work, semester control: credit  2nd semester: 18 h. – 9 lectures, 18 h. – 5 practical works, 6 h. – 3 consultations, 48 h. – independent work (including 20 h. - course work), semester control: combined exam |
|  | Schedule (terms) of study of the subject | First course, 1st and 2nd semester |
|  | Prerequisites for learning the discipline | In parallel, the disciplines "Theory of Algorithms", "Discrete Mathematics" and "Fundamentals of Programming" are studied. |
|  | Abstract (content) of the discipline | Mandatory discipline of professional and practical training, contains content modules:  1. OOP principles. Classes, objects, encapsulation  2. Inheritance and dynamic polymorphism.  3. Development of object-oriented software applications.  4. .Net platform and C # object model |
|  | Competencies, knowledge, skills, understanding that a higher education acquirer has in the learning process | Knowledge of general principles of object-oriented programming, implementation of object model in C ++ and C # programming languages; possession of generalized programming tools and a standard template library. |
|  | Learning outcomes of a Higher Education applicant | Creativity, ability to object-oriented thinking, knowledge of object-oriented programming languages ​​and the ability to apply an object-oriented approach when designing complex software systems. Ability to develop object-oriented graphics applications. |
|  | Assessment system in accordance with each task for taking tests/exams | 1st semester:  1. Work out and defend practical classes.  2. Pass 2 computer tests (10 questions on the analysis of program code for 12 minutes)  3. Perform Individual homework  4. Get at least 60 points per semester.  Semester grade (credit) : (3-5)х2 tests +(7-12)х6 practical works +(12-18)х ividual homework=(60-100) points.  2nd semester:  1. Work out and defend practical classes.  2. Pass 2 computer tests (10 questions on the analysis of program code for 12 minutes)  3. Perform Individual homework  4. Perform and defend a term paper (60-100 points).  5. Get at least 60 points per semester.  6. Pass the combined exam.  Semester grade : (3-5)х2 tests +(7-15)х5 practical works +(8-15)х ividual homework = (60-100) points.  Exam score  =(60-100) points.  The exam is combined in the form of a computer. test and practical task (test of 20 tasks, duration of 30 minutes, practical task - coding of a simple object-oriented application). |
|  | The quality of the educational process | Adherence to the principles of academic integrity (http://lib.nure.ua/plagiat). Update of the working program of the discipline - 2019. Practical classes are performed in the laboratory of the department, software - Visual Studio 2019 |
|  | Methodological support | 1. Complex of educational and methodical support of the educational discipline "Object-oriented programming" the first (bachelor's) level of higher education specialty 126 - Information systems and technologies [Electronic resource] / KNURE; developed. T.G. Bilova. - Kharkiv, 2018. - 301 p.<http://catalogue.nure.ua/knmz> |
|  | The developer of the Syllabus | T.G. Bilova, Associate Professor of Informational Control Systems Department, Candidate of Technical Sciences, Associate Professor  Е-mail: [tetiana.bilova@nure.ua](mailto:tetiana.bilova@nure.ua) |