|  |
| --- |
| Discipline „Distributed Processing in Internet”, code: 42 |
| Annotation:  The course aims to develop the students' knowledge for the development of distributed applications. The main concepts and problems in the development of distributed programming systems are discussed. There are presented basic approaches and up-to-date technologies for building distributed applications, as well as specific programming tools for their implementation. The discipline covers topics in middleware, server technologies, remote access communication protocols. Special attention is paid to object-oriented distributed decision technologies. |
| Main issues of the syllabus content:   * Distributed systems. Principles and functioning. Characteristics of distributed systems. * Processes and threads. Principles and implementation of threads. Threaded libraries. Pthread library for Linux. Models of multi-threaded programs. Thread communication and Sync. Windows threads implementation. * Classes for working with threads in object-oriented languages Java, C #. * Basic programming models for distributed systems. Client-server model. Client-server model implementation. Features. Server architecture. * Instrumental environments for realization of the client-server model. Medium layer. * Sockets. Interaction through sockets. Socket types. System calls for sockets. Setting the address information. * Multiplexed service via sockets. Sockets for group communication. * Socket classes in object-oriented programming languages. Java sockets. Features of implementation. Sockets in .Net. * Remote Procedure Call - RPC. Program model. Sending parameters. Error semantics. Dynamic binding. Formal specification of RPC server. * Implementation of RPC in Linux. Sun RPC. Server specification. Stub generator. RPC server and client development technology. * Distributed objects. Remote Calling Methods. Middleware systems based on remote call methods. Example – CORBA, DCOM. * Remote Calling (Java RMI). Remote interfaces. Create stub and skeletons. Using Java RMI to develop distributed applications. * Messaging-based systems. MPI. Features of implementation |