Innovative ICT Education for Social-Economic Development (IESED)
574283-EPP-1-2016-1-LT-EPPKA2-CBHE-JP

PROGRAMMING TECHNOLOGIES

Minsk 2017

PLAN OF EDUCATIONAL DISCIPLINE

Year of study	Semester	Academic hours					Hours of course work	ECTS	Number of hours
		Total	Lectures	Lab	Practice/ seminar	Independent work			
1	5,6	190	48	48		94		7	Full-time
2	5,6	190	16	12		162		7	Part-time

1. COMPETENCIES

1. Analyze perspectives and directions of development of information systems and technologies
2. Perform modeling, design of software tools and documentation to support activities in various subject areas

2. COURSE GOAL

Training of a specialist with fundamental knowledge and practical skills in the design of software systems; technologies of programming, implemented in modern frameworks

3. COURSE OUTCOMES

After completing this course student will be able to:

- analyze the subject area and choose the architecture of a software application, technologies and frameworks for solving specific problems;
- develop applications using various programming technologies.
- explain the architectural principles of building a software application.
- explain the principles of basic programming technologies and their implementation in modern frameworks.

4. EDUCATIONAL AND METHODOLOGICAL MAP (for full-time education)

Section number	Topics, classes; list of issues to be studied	Number of academic hours				Form of knowledge control
		Total	Lecture	laboratory classes	Independent work	
1.	Software systems (SS) development methodology	26	10	16		Final examination
1.1.	SS life cycle.	1	1			
1.2.	SS development process of (methods, tools).	6	2	4		Laboratory work
1.3.	SS development methodologies (MSF, RUP, AGILE).	1	1			
1.4.	SS design processes (domain, requirements)	12	4	8		Laboratory work
1.5.	Organization of teamwork.	5	1	4		Laboratory work
1.6.	Object-oriented development features.	1	1			
2.	Basics of designing SS	19	9	10		Final examination
2.1.	Architecture of SS.	1	1			
2.2.	Model View Controller architecture.	6	2	4		Laboratory work

2.3.	Model View View-Model architecture.	1	1			
2.4.	Service-oriented architecture: SOAP, REST.	6	2	4		Laboratory work
2.5.	Microservice architecture.	2	2			
2.6.	General principles of designing application classes.	3	1	2		
3.	Component programming	28	12	16		Final examination
3.1.	General principles.	1	1			
3.2.	Using interfaces.	1	1			
3.3.	Dependency injection.	2	2			
3.4.	Implementation of component programming in .NET.	12	4	8		Laboratory work
3.5.	Implementation of component programming in Java EE.	12	4	8		Laboratory work
4.	Event-oriented programming	27	9	18		Final examination
4.1.	General principles of constructing an event-oriented application.	1	1			
4.2.	Events, listeners, events providers.	1	1			
4.3.	Synchronous and asynchronous event processing.	1	1			
4.4.	Event-driving programming in VCL	8	2	6		Laboratory work
4.5.	Event- driving programming in Java EE.	8	2	6		Laboratory work
4.6.	Event- driving programming in.NET.	8	2	6		Laboratory work
5.	Technologies of data warehouses access.	17	5	12		Final examination
5.1.	Object-Relational Mapping	1	1			
5.2.	Implementation of ORM in Java EE.	8	2	6		Laboratory work
5.3.	Implementation of ORM in .NET.	8	2	6		Laboratory work
6.	Web Application Development Technologies	73	21	52		Final examination
6.1.	JEE Servlets.	5	1	4		Laboratory work
6.2.	Platform Spring MVC.	16	4	12		Laboratory work
6.3.	The ASP.NET platform.	16	4	12		Laboratory work
6.4.	Java Server Pages.	8	2	6		Laboratory work
6.5.	Java Server Faces.	10	4	6		Laboratory work
6.6.	Active Server Pages.	10	4	6		Laboratory work
6.7.	REST service development.	8	2	6		Laboratory work
	TOTAL	190	66	124		

5. EDUCATIONAL AND METHODOLOGICAL MAP (for part-time education)

Section number	Topics, classes; list of issues to be studied	Number of academic hours				Form of knowledge control
		Total	Lecture	laboratory classes	Independent work	
1.	Software systems (SS) development methodology	26	1	0	25	Final examination

1.1.	SS life cycle.	1	0,1		0,9	
1.2.	SS development process of (methods, tools).	6	0,1		5,9	Independent work
1.3.	SS development methodologies (MSF, RUP, AGILE).	1	0,2		0,8	
1.4.	SS design processes (domain, requirements)	12	0,2		11,8	Independent work
1.5.	Organization of teamwork.	5	0,2		4,8	Independent work
1.6.	Object-oriented development features.	1	0,2		0,8	
2.	Basics of designing SS	19	2	0	17	Final examination
2.1.	Architecture of SS.	1	0,2		0,8	
2.2.	Model View Controller architecture.	6	0,4		5,6	Independent work
2.3.	Model View View-Model architecture.	1	0,4		0,6	
2.4.	Service-oriented architecture: SOAP, REST.	6	0,4		5,6	Independent work
2.5.	Microservice architecture.	2	0,4		1,6	
2.6.	General principles of designing application classes.	3	0,2		2,8	
3.	Component programming	28	2	2	24	Final examination
3.1.	General principles.	1	0,2		0,8	
3.2.	Using interfaces.	1	0,2		0,8	
3.3.	Dependency injection.	2	0,4		1,6	
3.4.	Implementation of component programming in .NET.	12	0,6	1	10,4	Laboratory work
3.5.	Implementation of component programming in Java EE.	12	0,6	1	10,4	Laboratory work
4.	Event-oriented programming	27	2	2	23	Final examination
4.1.	General principles of constructing an event-oriented application.	1	0,2		0,8	
4.2.	Events, listeners, events providers.	1	0,2		0,8	
4.3.	Synchronous and asynchronous event processing.	1	0,2		0,8	
4.4.	Event-driving programming in VCL	8	0,2		7,8	Independent work
4.5.	Event- driving programming in Java EE.	8	0,6	1	6,4	Laboratory work
4.6.	Event- driving programming in.NET.	8	0,6	1	6,4	Laboratory work
5.	Technologies of data warehouses access.	17	2	2	13	Final examination
5.1.	Object-Relational Mapping	1	0,2		0,8	
5.2.	Implementation of ORM in Java EE.	8	0,9	1	6,1	Laboratory work
5.3.	Implementation of ORM in .NET.	8	0,9	1	6,1	Laboratory work
6.	Web Application Development Technologies	73	7	6	60	Final examination
6.1.	JEE Servlets.	5	0,2		4,8	Independent work
6.2.	Platform Spring MVC.	16	2,5	2	11,5	Laboratory work
6.3.	The ASP.NET platform.	16	2,5	2	11,5	Laboratory work
6.4.	Java Server Pages.	8	0,5	1	6,5	Laboratory work
6.5.	Java Server Faces.	10	0,5		9,5	Independent work
6.6.	Active Server Pages.	10	0,5		9,5	Independent work
6.7.	REST service development.	8	0,3	1	6,7	Independent work
	TOTAL	190	16	12	162	

6. THEORETICAL CONTENT

№	Topic number	Content
1.	**Software systems (SS) development methodology**	
1.1.	SS life cycle.	Life cycle models. Technological processes.
1.2.	SS development process of (methods, tools).	Types of notations. Computer Aided Software Engendering tools. Integrated Developer Environments.
1.3.	SS development methodologies (MSF, RUP, AGILE).	Microsoft Solution Framework. Rational Unified Process. Agile software development. Extreme Programming.
1.4.	SS design processes (domain, requirements)	Analysis of the subject area and software requirements. Requirements analysis. The model of application classes.
1.5.	Organization of teamwork.	Functions of the project participants. Tools and services for organization of team work. Integrating with IDEs.
1.6.	Object-oriented development features.	Iterative development in object-oriented design. Criteria for selecting classes for development at the next iteration.
2.	**Basics of designing SS**	
2.1.	Architecture of SS.	Software architecture. Data flow. Call-return. Interactive systems. Systems based on the data warehouse. Domain Driven Design.
2.2.	Model View Controller architecture.	Class structure. Class groups. Scenario for processing the user action. Scenario of system initialization. The main steps of implementation.
2.3.	Model View View-Model architecture.	Two-way communication with the presentation. Model-View-Presenter. Reactive Programming
2.4.	Service-oriented architecture: SOAP, REST.	Simple Object Access Protocol. Representational State Transfer. REST Architecture Requirements.
2.5.	Microservice architecture.	Services in a microservice architecture. continuous delivery software development process. Decentralized Data Management. Authorization and authentication in microservice system.
2.6.	General principles of designing application classes.	Basic principles of design. Mechanisms of reuse. End-to-end functionality. Designing for future changes.
3.	**Component programming**	
3.1.	General principles.	The concept of the software component. Deployment Components. Component framework. Basic services. Component model.
3.2.	Using interfaces.	Using interfaces to make components framework.
3.3.	Dependency injection.	Basics principals of dependency injection. Inversion of Control. Types of dependency injection. Scope.
3.4.	Implementation of component programming in .NET.	Ways to implement a synchronous and asynchronous way of processing messages.
3.5.	Implementation of component programming in Java EE.	Enterprise Java Beans. Bean class requirements. Dependency injection implementation. Java Naming and Directory Interface.
4.	**Event-oriented programming**	
4.1.	General principles of constructing an event-oriented application.	Models of messaging. Messages queue. Message processing loop. Processing of messages. Message-Oriented Middleware.
4.2.	Events, listeners, events providers.	The main participants in the messaging process and their functions.
4.3.	Synchronous and asynchronous event processing.	Synchronous and Asynchronous Processing.
4.4.	Event-driving programming in VCL	Class hierarchy VCL. Processing of messages in VCL. Message types. Exception classes.

№	Topic number	Content
4.5.	Event- driving programming in Java EE.	Java Message system. Basic Interfaces. Types of messages. QueueSender, TopicPublisher. QueueReceiver, TopicSubscriber. Creating a session.
4.6.	Event- driving programming in.NET.	
5.	**Technologies of data warehouses access.**	
5.1.	Object-Relational Mapping	Basics principals. Entity-classes. Ways of Mapping describing.
5.2.	Implementation of ORM in Java EE.	Java Persistence. Persistence context. Configuring the system. Hibernate. Hibernate Query Language.
5.3.	Implementation of ORM in .NET.	
6.	**Web Application Development Technologies**	
6.1.	JEE Servlets.	Servlet container. Types of servlets. Servlet interface. The HttpServlet class. Servlet registration. Interface HttpSession. Methods of client authentication. WAR file.
6.2.	Platform Spring MVC.	Spring framework overview. Dependency injection implementation. Web-application configuration. Main components of web-application and their configuration. Request life cycle. Request mapping.
6.3.	The ASP.NET platform.	
6.4.	Java Server Pages.	JSP – Architecture. Life cycle of a JSP Page. Elements of JSP. JSTL. User-defined tags.
6.5.	Java Server Faces.	JSF – Architecture. Request lifecycle. Managed beans. Views. Main tags. Facelets. Special components.
6.6.	Active Server Pages.	
6.7.	REST service development.	Making REST-controllers in Spring. Custom response formation. JSON processor for Java.

7. LABORATORY PRACTICE

№	Topic number	Content
1.	**Software systems (SS) development methodology**	
1.2.	SS development methodologies (MSF, RUP, AGILE).	Configuring and customization Eclipse IDE. Configuring and customization Visual Studio IDE.
1.4.	SS design processes (domain, requirements)	Designing an information system, designing a domain model, developing requirements
1.5.	Organization of teamwork.	Using Git-based web-services for teamwork organization.
2.	**Basics of designing SS**	
2.2.	Model View Controller architecture.	Developing application class model based on the MVC architecture.
2.4.	Service-oriented architecture: SOAP, REST.	Developing application class model based on the Service-oriented architecture.
2.6.	General principles of designing application classes.	Using OOP principles when designing an application.
3.	**Component programming**	
3.4.	Implementation of component programming in .NET.	Using component framework to develop an application
3.5.	Implementation of component programming in Java EE.	Using component framework to develop an application
4.	**Event-oriented programming**	
4.4.	Event-driving programming in VCL	Developing of event-driving VCL visual component.
4.5.	Event- driving programming in Java EE.	Using event-driven beans.
4.6.	Event- driving programming in.NET.	

№	Topic number	Content
5.	**Technologies of data warehouses access.**	
5.2.	Implementation of ORM in Java EE.	Organization of access to the relational database using Hibernate for the information system project
5.3.	Implementation of ORM in .NET.	
6.	**Web Application Development Technologies**	
6.1.	JEE Servlets.	Creating servlet based web-application
6.2.	Platform Spring MVC.	Developing component-based web-application in spring framework
6.3.	The ASP.NET platform.	
6.4.	Java Server Pages.	Developing view for web-application
6.5.	Java Server Faces.	Using JSF framework to create GUI of web-application
6.6.	Active Server Pages.	
6.7.	REST service development.	Creating REST API for information system.

8. ASSIGNMENT FOR INDEPENDENT WORK

№	Tasks
1.	Design and develop a software system on any of the platforms presented in the course. Choose domain independently (for example: the learning progress of the student group, personal game statistics, personal expenses, etc.)
2.	Create front-end application on base View View-Model architecture.
3.	Organize own developers team on bitbuket or github platform.
4.	Developing own JSP tag.
5.	Create web-application views on base of Facelets.

9. SYSTEM OF ASSESSMENT OF KNOWLEDGE AND SKILLS (ACCORDING TO THE NATIONAL REQUIREMENTS)

The evaluation criteria of the results on aten-pointscale.

A ten-point scale, depending on the grade and the mark, includes the following criteria:

10 (ten) points, passed:

- systematized, deep and full knowledge on all sections of the curriculum of the institution of higher education in the academic discipline, as well as on major issues that go beyond its limits;
- accurate use of scientific terminology (including in a foreign language), competent, logically correct statement of the answer to questions;
- perfect mastering of the tools of the academic discipline, the ability to use it effectively in formulation and solution of scientific and professional problems;
- the expressed ability independently and creatively to solve complex problems in non-standard situations;
- complete and profound studying of basic, additional literature on the subject of the discipline;
- the ability to freely navigate in theories, concepts and directions on the discipline and give them an analytical assessment, use the scientific achievements of other disciplines;
- creative independent work on practical, laboratory classes, active creative participation in group discussions, high level of the culture of performance of tasks.

9 (nine) points, passed:

- systematized, deep and full knowledge on all sections of the curriculum of the institution of higher education on the academic discipline;
- accurate use of scientific terminology (including in a foreign language), competent, logically correct statement of the answer to questions;

- mastering of the tools of the academic discipline, the ability to use it effectively in formulation and solution of scientific and professional problems;
- ability independently and creatively to solve complex problems in non-standard situations within the curriculum of the institution of higher education on the academic discipline;
- complete studying of basic, additional literature on the subject of the discipline, recommended by the curriculum of the institution of higher education on the discipline;
- the ability to navigate in theories, concepts and directions on the discipline and give them an analytical assessment;
- Systematic, active independent work on practical, laboratory classes, active creative participation in group discussions, high level of the culture of performance of tasks.

8 (eight) points, passed:
- systematized, deep and full knowledge on all sections of the curriculum of the institution of higher education in the academic discipline in the volume of the curriculum of the institution of higher education on the discipline;
- use of scientific terminology (including in a foreign language), competent, logically correct statement of the answer to questions, the ability to make sound conclusions and generalizations;
- mastering of the tools of the academic discipline (methods of complex analysis, information technology), the ability to use it effectively in formulation and solution of scientific and professional problems;
- ability independently to solve complex problems within the curriculum of the institution of higher education on the academic discipline;
- studying of basic, additional literature, recommended by the curriculum of the institution of higher education on the discipline;
- the ability to navigate in theories, concepts and directions on the discipline and give them an analytical assessment;
- active independent work on practical, laboratory classes, systematic participation in group discussions, high level of the culture of performance of tasks.

7 (seven) points, passed:
- systematized, deep and full knowledge on all sections of the curriculum of the institution of higher education on the academic discipline;
- use of scientific terminology (including in a foreign language), competent, logically correct statement of the answer to questions, the ability to make sound conclusions and generalizations;
- mastering of the tools of the academic discipline, the ability to use it effectively in formulation and solution of scientific and professional problems;
- free possession of generic solutions within the curriculum of the institution of higher education on the academic discipline;
- studying of basic, additional literature, recommended by the curriculum of the institution of higher education on the discipline;
- the ability to navigate in basic theories, concepts and directions on the discipline and give them an analytical assessment;
- independent work on practical, laboratory classes, participation in group discussions, high level of the culture of performance of tasks.

6 (six) points, passed:
- sufficiently fulland systematized knowledge in the volume of the curriculum of the institution of higher education on the discipline;
- use of the necessary scientific terminology, competent, logically correct statement of the answer to questions, the ability to make sound conclusions and generalizations;

- mastering of the tools of the academic discipline, the ability to use it effectively in solution of scientific and professional problems;
- ability independently to apply generic solutions within the curriculum of the institution of higher education on the academic discipline;
- studying of basic literature, recommended by the curriculum of the institution of higher education on the discipline;
- the ability to navigate in basic theories, concepts and directions on the discipline and give them acomparative assessment;
- active independent work on practical, laboratory classes, periodic participation in group discussions, high level of the culture of performance of tasks.

5 (five) points, passed:
- sufficient knowledge in the volume of the curriculum of the institution of higher education on the discipline;
- use of scientific terminology, competent, logically correct statement of the answer to questions, the ability to make sound conclusions;
- mastering of the tools of the academic discipline, the ability to use it in solution of scientific and professional problems;
- ability independently to apply generic solutions within the curriculum of the institution of higher education on the academic discipline;
- studying of basic literature, recommended by the curriculum of the institution of higher education on the discipline;
- the ability to navigate in basic theories, concepts and directions on the discipline and give them a comparative assessment;
- active independent work on practical, laboratory classes, periodic participation in group discussions, high level of the culture of performance of tasks;
- independent work on practical, laboratory classes, periodic participation in group discussions, sufficient level of the culture of performance of tasks.

4 (four) points, passed:
- sufficient knowledge within the educational standard of higher education;
- studying of basic literature, recommended by the curriculum of the institution of higher education on the discipline;
- use of scientific terminology, logical statement of the answer to questions, the ability to make sound conclusions;
- ability to draw conclusions without essential errors;
- mastering of the tools of the academic discipline, the ability to use it in solution of standard (typical) tasks;
- ability to solve standard (typical) tasks under the guidance of a teacher;
- ability to navigate in basic theories, concepts and directions on the discipline and give them an assessment;
- work under the guidance of a teacher on practical, laboratory classes, the permissible level of the culture of performance of tasks.

3 (three) points, failed:
- insufficient knowledge within the educational standard of higher education;
- studying of basic literature, recommended by the curriculum of the institution of higher education on the discipline;
- knowledge of a part of the basic literature, recommended by the curriculum of the institution of higher education on the discipline;

- use of scientific terminology, presentation of answers to questions with significant, logical errors;
- weak possession of the tools of the academic discipline, incompetence in solving standard (typical) tasks;
- inability to navigate in basic theories, concepts and directions on the discipline;
- work under the guidance of a teacher on practical, laboratory classes, the permissible level of the culture of performance of tasks.
- passivity on practical, laboratory classes, law level of the culture of performance of tasks.

2 (two) points, failed:
- fragmented knowledge within the educational standard of higher education;
- knowledge of individual literary sources, recommended by the curriculum of the institution of higher education on the discipline;
- inability to use scientific terminology of the academic discipline, the presence in the answer rude, logical errors;
- passivity on practical, laboratory classes, law level of the culture of performance of tasks.

1 (one) point, failed:
- lack of knowledge and (competences) within the educational standard of higher education, failure to answer, failure to appear for attestation without good cause.

10. METHODS AND MEANS OF IMPLEMENTATION OF THE CONTENT OF THE EDUCATIONAL PROGRAM AND TRAINING OF EDUCATIONAL, TRAINING AND METHODICAL MATERIALS

The training will be conducted using classical methods, project methods and distance learning technologies, implemented on the portal SDO.VSTU.BY. Trainees will be provided with access to the SDO.VSTU.BY portal.
In the classroom, students will learn the discipline directly in the computer class. The following software (software) will be used during the training:

- IDE MS Visual Studio Community,
- IDE NetBeans.

The following tools and technologies were used in the preparation of training, teaching and methodological materials:
- Adobe PDF.
- LMS Moodle.
- MS Visual Studio Community 2015.
- NetBeans 8.1.
- Programs for recording and editing video.

11. THE INFORMATION-METHODOLOGICAL PART

Main literature

№	Authors	Bibliographic description
1.	Subrahmanyam Allama-raju, Andrew Longshaw, Daniel O'Connor, more 6	Java Server Programming J2EE 1.4 Edition - New Delhi: Dreamtech, 2010.
2.	Keogh, J.	J2EE: The Complete Reference New Delhi: Tata McGraw Hill Education Private Limited, 2010
3.	Б. Я. Советов, В. В. Цехановский	Информационные технологии: учебник для студентов вузов, обучающихся по направлениям подготовки дипломированных

		специалистов "Информатика и вычислительная техника" и "Информационные системы" - 4-е изд., стер. - Москва: Высшая школа, 2008. - 263 с.
4.	сост. Н. М. Прибыльская	Программирование сетевых приложений: лабораторные работы (практикум) для студентов спец. 1-40 01 01 "Программное обеспечение информационных технологий" и 1-40 01 02 "Информационные системы и технологии" / БНТУ;. - Минск: БНТУ, 2013.
5.	Архангельский А.Я.	Компоненты C++ Builder. Справочное и методическое пособие. - М.: Бином-Пресс, 2008.

Additional literature

6.	Кулямин В.В.	Технологии программирования. Компонентный подход. - М: ИНТУИТ-Бином, 2007. - 463 с.
7.		Материалы сайта javaee.github.io
8.		Материалы сайта docs.microsoft.com/ru-ru/dotnet/
9.		Материалы сайта nodejs.org