

**МИНОБРНАУКИ РОССИИ**  
**Ярославский государственный университет им. П.Г. Демидова**

Кафедра вычислительных и программных систем

УТВЕРЖДАЮ

Декан факультета ИВТ

 Д.Ю. Чалый

« 23 » мая 2023 г.

**Рабочая программа дисциплины**  
«Программирование в Windows и в сетях Windows»

**Направление подготовки**  
01.03.02 Прикладная математика и информатика

**Направленность (профиль)**  
«Программирование и технологии искусственного интеллекта»

**Квалификация выпускника**  
Бакалавр

**Форма обучения**  
очная

Программа рассмотрена на  
заседании кафедры  
от 21 апреля 2023 г.,  
протокол № 8

Программа одобрена НМК  
факультета ИВТ  
протокол № 6 от  
28 апреля 2023 г.

Ярославль

### 1. Цели освоения дисциплины

Целями дисциплины «Программирование в Windows и в сетях Windows» являются ознакомление студентов с современными средствами разработки Windows-приложений, а именно, со средой Microsoft Visual Studio и библиотекой Microsoft Foundation Classes, развитие понимания основных механизмов функционирования операционной системы Windows, способов и средств взаимодействия с ними, а также ознакомление студентов со средствами организации взаимодействия и передачи информации между приложениями в сетях Windows, используемых для этого протоколов и интерфейсов, способах синхронизации и повышения эффективности сетевого взаимодействия.

### 2. Место дисциплины в структуре ОП бакалавриата

Дисциплина «Программирование в Windows и в сетях Windows» относится к вариативной части (дисциплина по выбору) ОП бакалавриата.

Для изучения данной дисциплины студенту необходимо знать язык программирования C++, иметь навыки объектно-ориентированного программирования и начальные знания об операционной системе Windows.

Знания и навыки, полученные при изучении данной дисциплины, используются учащимися при изучении других дисциплин компьютерного цикла, а также при подготовке курсовых и дипломных работ.

### 3. Планируемые результаты обучения по дисциплине, соотнесенные с планируемыми результатами освоения ОП бакалавриата

Процесс изучения дисциплины направлен на формирование следующих элементов компетенций в соответствии с ФГОС ВО, ОП ВО и приобретения следующих знаний, умений, навыков и (или) опыта деятельности:

Формируемая компетенция (код и формулировка)	Индикатор достижения компетенции (код и формулировка)	Перечень планируемых результатов обучения
<b>Профессиональные компетенции</b>		
ПК-2 Способен к разработке и применению алгоритмических и программных решений в области системного и прикладного программного обеспечения, в том числе методы машинного обучения.	ПК – 2.1 Демонстрирует знания в области современных языков и технологий программирования, комплексов прикладных компьютерных программ.	<b>Знать:</b> – современные языки программирования и средства разработки приложений для ОС Windows. <b>Уметь:</b> – выбрать для решения конкретной задачи подходящий язык, среду разработки и библиотечные средства. <b>Владеть навыками:</b> – поиска актуальной информации по языкам и средствам программирования в сети "Интернет".

### 4. Объем, структура и содержание дисциплины

Общая трудоемкость дисциплины составляет 5 зач. ед., 180 акад. час.

№ п/п	Темы (разделы) дисциплины, их содержание	Семестр	Виды учебных занятий, включая самостоятельную работу студентов, и их трудоемкость (в академических часах)						Формы текущего контроля успеваемости  Форма промежуточной аттестации
			Контактная работа						
			лекции	практические	лабораторные	консультации	аттестационные испытания	самостоятельная работа	
1.	Установка среды Visual Studio и справочной системы. Создание программ в среде Visual Studio	5	1		1				
2.	Модель программирования в Windows с использованием графического интерфейса	5	1		1				
3.	Создание программ с графическим интерфейсом	5	1		1			2	
4.	Реализация представления	5	1		1	1		2	
5.	Реализация документа	5	1		1	1		2	
6.	Хранение данных	5	1		1			2	сдача проекта
7.	Прокрутка и разделение окон представления	5	1		1			2	
8.	Перемещаемые панели и строки состояния	5	1		1			2	
9.	Создание диалоговых окон	5	1		1			2	
10.	Разработка диалоговых приложений	5	1		1	1		2	
11.	Создание многодокументных приложений	5	1		1			2	
12.	Ввод/вывод символов	5	1		1			2	
13.	Использование функций рисования	5	1		1			2	
14.	Растровые изображения и битовые операции	5	1		1			3	сдача проекта
15.	Печать и предварительный просмотр	5	1		1			4	
16.	Многопоточные приложения	5	2		2	1		2,7	
17.	Связи между процессами	5	1		1				Экзамен
	<b>Всего за 5 семестр</b>		<b>18</b>		<b>18</b>	<b>4</b>		<b>31,7</b>	<b>Зачет</b>
18.	Механизм OLE	6			2			3	
19.	Создание и применение элементов ActiveX	6			2			3	
20.	Динамически подключаемые библиотеки	6			2			3	
21.	Интерфейс NetBIOS	6			6			3	
22.	Перенаправитель.	6			2			3	

	Почтовые ящики								
23.	Именованные каналы	6			2	1		3	
24.	Сетевые протоколы	6			2			3	
25.	Семейства адресов и разрешение имен	6			2	1		3	
26.	Основы интерфейса Winsock	6			2	1		3	
27.	Ввод-вывод в Winsock	6			12	2		3	сдача проекта
28.	Параметры сокета и команды управления вводом-выводом	6			2			1	
	<b>Всего за 6 семестр</b>				<b>36</b>	<b>5</b>		<b>31</b>	<b>Экзамен</b>
	<b>Всего</b>		<b>18</b>		<b>54</b>	<b>9</b>		<b>62,7</b>	

Для самостоятельной работы используются задания из учебных пособий [3] и [5]. Они доступны в электронной форме в сети факультета. Распределение их по темам соответствует темам лекций.

**Зачет** проходит в форме разработки и защиты самостоятельного проекта, разработанного с использованием библиотеки MFC.

**Экзамен** состоит из двух частей. В практической части студент должен разработать и сдать сетевое приложение в соответствии с требованиями, выложенными в сети факультета. В теоретической он должен ответить на вопросы по темам, приведенным в приложении.

#### Содержание разделов дисциплины:

##### 1. Установка среды Visual Studio и справочной системы. Создание программ в среде Visual Studio

Установка Visual Studio и справочной системы MSDN.

Процесс компиляции и сборки программы. Создание проекта программы. Промежуточные файлы Visual Studio. Компоненты Visual Studio. Создание и редактирование исходных файлов программы. Изменение установок проекта. Построение программы. Выполнение программы. Отладка программы.

##### 2. Модель программирования в Windows с использованием графического интерфейса

Обработка сообщений. Интерфейс графического устройства. Программирование, основанное на ресурсах. Управление памятью. Динамически подключаемые библиотеки. Интерфейс прикладных программ. Win32 API.

##### 3. Создание программ с графическим интерфейсом

Генерация исходного кода. Редактирование исходного кода. Построение и запуск программы. Классы и файлы программы. Этапы выполнения программы. Исключения в библиотеке MFC.

##### 4. Реализация представления

Реализация графического представления. Реализация текстового представления.

##### 5. Реализация документа

Сохранение графических данных. Перерисовка окна. Добавление команд в меню. Удаление данных документа. Реализация команд меню.

## **6. *Хранение данных***

Принципы сохранения и загрузки данных документа из файлов на диске. Реализация стандартных команд меню File: New, Open, Save, Save As. Реализация технологии drag-and-drop.

## **7. *Прокрутка и разделение окон представления***

Добавление средств прокрутки окна. Добавление средств разделения окна. Обновление окна представления. Эффективная перерисовка.

## **8. *Перемещаемые панели и строки состояния***

Добавление в новую программу перемещаемой панели инструментов и строки состояния в процессе генерации кода. Добавление перемещаемой панели инструментов в существующую программу. Добавление строки состояния в существующую программу.

## **9. *Создание диалоговых окон***

Создание модальных диалоговых окон. Создание немодальных диалоговых окон. Создание диалоговых окон со вкладками. Диалоговые окна общего назначения.

## **10. *Разработка диалоговых приложений***

Простые диалоговые программы. Программы просмотра форм.

## **11. *Создание многодокументных приложений***

Многодокументный интерфейс. Создание MDI-программы в среде Developer Studio. Основные классы MDI-программы. Настройка ресурсов.

## **12. *Ввод/вывод символов***

Отображение текста. Чтение кодов символов, вводимых с клавиатуры. Управление курсором при редактировании.

## **13. *Использование функций рисования***

Создание объекта контекста устройства. Выбор средств рисования внутри объекта. Установка атрибутов рисования для объекта. Создание графических изображений. Функции рисования - члены класса CDC.

## **14. *Растровые изображения и битовые операции***

Создание растровых изображений. Выполнение битовых операций при отображении. Отображение значков.

## **15. *Печать и предварительный просмотр***

Добавление в программу средств печати и предварительного просмотра. Усовершенствованная печать.

## **16. *Многопоточные приложения***

Создание и управление вторичными потоками. Особенности использования MFC-классов в многопоточных программах. Синхронизация потоков.

## **17. *Связи между процессами***

Запуск новых процессов. Синхронизация процессов. Обмен данными по каналам. Совместное использование памяти. Использование буфера обмена.

## **18. *Механизм OLE***

Внедрение, связывание и автоматизация. Разработка программы-сервера. Разработка программы-контейнера.

### **19. Создание и применение элементов ActiveX**

Разработка элементов ActiveX. Определение свойств, методов и событий элементов ActiveX. Разработка программы-контейнера элементов ActiveX.

### **20. Динамически подключаемые библиотеки**

Основы DLL. Экспорт и импорт функций. DLL-расширения и обычные DLL.

### **21. Интерфейс NetBIOS**

NetBIOS и сетевая модель OSI. Интерфейс Microsoft NetBIOS. Основы программирования NetBIOS. Типовые процедуры NetBIOS. Дейтаграммные операции. Дополнительные команды NetBIOS.

### **22. Перенаправитель. Почтовые ящики**

Универсальные правила именования. Компоненты сетевого доступа. Протокол SMB. Имена почтовых ящиков. Использование архитектуры клиент-сервер. Дополнительные API-функции почтовых ящиков.

### **23. Именованные каналы**

Правила именования каналов. Режимы передачи. Простой сервер и клиент. Усовершенствованный сервер каналов.

### **24. Сетевые протоколы**

Интерфейс прикладного программирования Winsock. Характеристики протоколов. Сетевые протоколы, поддерживаемые Win32. Работа с Winsock.

### **25. Семейства адресов и разрешение имен**

Протокол IP. Протоколы NetBIOS. Протоколы IPX/SPX. Инфракрасные сокет. Протокол ATM. Дополнительные функции Winsock 2.

### **26. Основы интерфейса Winsock**

Инициализация Winsock. Проверка и обработка ошибок. Протоколы с установлением соединения: основные операции и соответствующие функции Win32 API. Протоколы, не требующие соединения.

### **27. Ввод-вывод в Winsock**

Режимы работы сокетов. Модели управления вводом-выводом сокетов: select, WSAAsyncSelect, WSAEventSelect, модель перекрытого ввода-вывода, модель портов завершения.

### **28. Параметры сокета и команды управления вводом-выводом**

Параметры сокета, их уровни. Функции ioctlsocket и WSAIoctl.

## **5. Образовательные технологии, используемые при осуществлении образовательного процесса по дисциплине**

В процессе обучения используются следующие образовательные технологии:

**Вводная лекция** – дает первое целостное представление о дисциплине и ориентирует студента в системе изучения данной дисциплины. Студенты знакомятся с назначением и задачами курса, его ролью и местом в системе учебных дисциплин и в системе подготовки в целом. Дается краткий обзор курса, история развития науки и практики, достижения в этой сфере, имена известных ученых, излагаются перспективные направления исследований. На этой лекции высказываются методические и организационные особенности работы в рамках данной дисциплины, а также дается анализ рекомендуемой учебно-методической литературы.

**Академическая лекция** (или лекция общего курса) – последовательное изложение материала, осуществляемое преимущественно в виде монолога преподавателя. Требования к академической лекции: современный научный уровень и насыщенная информативность, убедительная аргументация, доступная и понятная речь, четкая структура и логика, наличие ярких примеров, научных доказательств, обоснований, фактов.

Лекции проводятся в компьютерном классе с использованием проектора для демонстрации презентаций и демонстрационных примеров. Такой подход позволяет разобрать достаточно большой объем материала и продемонстрировать весьма объемные примеры кода, что было бы невозможным при использовании традиционных средств. Лекции по времени совмещены с лабораторными занятиями, что дает возможность сразу закрепить прослушанный материал на практике и при необходимости получить консультации лектора.

**Лабораторное занятие** – занятие в компьютерном классе, посвященное освоению конкретных умений и навыков и закреплению полученных на лекции знаний.

Основной формой практической работы студентов по усвоению данного курса является выполнение ими самостоятельных проектов в рамках лабораторных (по каждой теме) и самостоятельных занятий по данному курсу. Все задания, заготовки проектов, демонстрационные и вспомогательные учебные материалы предоставляются учащимся в электронном виде и постоянно доступны в сети факультета. Это позволяет в случае пропуска занятия самостоятельно отработать пропущенную тему.

Промежуточная аттестация производится в форме отчетов студентов по самостоятельно выполненным проектам, окончательная аттестация в форме зачета и экзамена.

## **6. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень лицензионного программного обеспечения и информационных справочных систем (при необходимости)**

В процессе осуществления образовательного процесса используются:

- Windows 7 в качестве основной операционной системы;
- OfficeStd 2013 RUS OLP NL Acdmc 021-10232 для создания слайдов презентаций, используемых для чтения лекций;
- Интегрированная среда разработки Microsoft Visual Studio 2010 для подготовки демонстрационных примеров, а также заготовок лабораторных работ и учебных упражнений;
- OfficeStd 2013 RUS OLP NL Acdmc 021-10232 для формирования инструкций по выполнению учебных упражнений, а также текстов материалов для промежуточной и текущей аттестации;

## **7. Перечень основной и дополнительной учебной литературы, ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины**

а) основная литература:

1. Олифер В. Г. Компьютерные сети: принципы, технологии, протоколы.: учеб. пособие для вузов. / В. Олифер, Н. Олифер; М-во образования и науки РФ - 5-е изд. - СПб.: Питер, 2017. - 991 с.

б) дополнительная литература:

2. Васильчиков В.В. Разработка сетевых приложений для ОС Windows (практические примеры): учебное пособие. - Ярославль.: ЯрГУ, 2009.

3. Васильчиков В.В. Программирование в Visual C++ с использованием библиотеки MFC: учебное пособие. - Ярославль.: ЯрГУ, 2006.
4. Васильчиков В.В. Основы разработки сетевых Windows-приложений: учебное пособие. - Ярославль.: ЯрГУ, 2007.

в) программное обеспечение и Интернет-ресурсы:

- Visual Studio 2010.
- Информация по языкам программирования, операционным системам, примеры программ: [www.firststeps.ru](http://www.firststeps.ru), [infocity.kiev.ua](http://infocity.kiev.ua), [www.corp7.ivt.uniyar.ac.ru](http://www.corp7.ivt.uniyar.ac.ru)
- <https://msdn.microsoft.com/ru-ru/vstudio/default.aspx>
  - Электронно-библиотечная система «Юрайт»( <https://urait.ru/> ).
  - Электронно-библиотечная система «Лань»( <https://e.lanbook.com/>).

#### **8. Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине**

Материально-техническая база, необходимая для осуществления образовательного процесса по дисциплине включает в свой состав специальные помещения:

- учебные аудитории для проведения занятий лекционного типа и практических занятий (семинаров);
- учебные аудитории для проведения лабораторных занятий;
- учебные аудитории для проведения групповых и индивидуальных консультаций,
- учебные аудитории для проведения текущего контроля и промежуточной аттестации;
- помещения для самостоятельной работы;
- помещения для хранения и профилактического обслуживания технических средств обучения.

Специальные помещения укомплектованы средствами обучения, служащими для представления учебной информации большой аудитории.

Для проведения занятий лекционного типа предлагаются наборы демонстрационного оборудования и учебно-наглядных пособий, хранящиеся на электронных носителях и обеспечивающие тематические иллюстрации, соответствующие рабочим программам дисциплин.

Помещения для лабораторных занятий и самостоятельной работы обучающихся оснащены компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду организации.

Число посадочных мест в лекционной аудитории больше либо равно списочному составу потока, а в аудитории для практических занятий (семинаров) – списочному составу группы обучающихся.

#### **Автор(ы) :**

Зав. кафедрой

вычислительных и программных систем, к.т.н. \_\_\_\_\_ В.В. Васильчиков



**Приложение №1**  
**к рабочей программе дисциплины**  
**"Программирование в Windows и в сетях Windows"**

**Фонд оценочных средств**  
**для проведения текущей и промежуточной аттестации студентов**  
**по дисциплине**

**1. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций**

**1.1. Контрольные задания и иные материалы, используемые в процессе текущей аттестации**

**Задания для самостоятельной работы**

Для самостоятельной работы в пятом семестре используются учебные упражнения из учебного пособия [3]: В.В.Васильчиков "Программирование в Visual C++ с использованием библиотеки MFC: учебное пособие" - Ярославль: ЯрГУ, 2006.

Большинство из них являются обязательными для выполнения. Заготовки проектов, сценарии работы и пошаговые инструкции для выполнения каждого из упражнений выложены в локальной сети факультета. Упражнения заключаются в создании проекта приложения с графическим интерфейсом и последовательном развитии его функциональности. По завершении всех этапов развития приложения готовый проект сдается преподавателю. Для прохождения аттестации в течение семестра студенты должны выполнить все обязательные упражнения и сдать завершённый проект.

Для оценки качества сданного проекта применяются следующие критерии оценивания

<b>Оценка</b>	<b>Критерии</b>
Отлично	ОПК-3: Умеет спроектировать графический интерфейс и реализовать его с использованием подходящих элементов управления. Для добавления необходимой функциональности использует подходящие события и реализует их обработчики. Эффективно реализует обмен информацией между элементами управления и переменными программы. Понимает механизм динамического обмена данными. Программа выполняется для всех тестов ПК-7: Создает полноценное приложение с графическим интерфейсом. Поясняет код и изменяет его при необходимости
Хорошо	ОПК-3: Умеет спроектировать графический интерфейс и реализовать его с использованием подходящих элементов управления. Для добавления необходимой функциональности использует подходящие события и реализует их обработчики. Реализует обмен информацией между элементами управления и переменными программы. Программа выполняется для всех тестов, если какой-либо из тестов не выполняется программа дорабатывается. ПК-7: Создает приложение с графическим интерфейсом. Поясняет код и изменяет его при необходимости с небольшими неточностями

Удовлетворительно	<p>ОПК-3: В основном может спроектировать графический интерфейс и реализовать его с использованием подходящих элементов управления. Понимает связь между событием и обработчиком, с некоторыми подсказками может выбрать подходящее событие. Неэффективно организует обмен информацией между элементами управления и переменными программы. Программа выполняется для всех тестов, возможно кроме одного-двух крайних случаев.</p> <p>ПК-7: Создает приложение с графическим интерфейсом. С трудом поясняет код, не может изменить код при усложнении или существенном дополнении задачи</p>
Неудовлетворительно	<p>ОПК-3: Не может спроектировать графический интерфейс и реализовать его. Не понимает механизма событий. Программа не выполняется для большинства тестов.</p> <p>ПК-7: Не может создать приложение с графическим интерфейсом в среде разработки или создает неработающий проект. Не может пояснить код и изменить его.</p>

Для самостоятельной работы в шестом семестре используются учебные упражнения из учебного пособия [5]: В.В.Васильчиков "Разработка сетевых приложений для ОС Windows (практические примеры): учебное пособие" - Ярославль: ЯрГУ, 2009.

Большинство из них являются обязательными для выполнения. Заготовки проектов, сценарии работы и пошаговые инструкции для выполнения каждого из упражнений выложены в локальной сети факультета. Упражнения заключаются в создании проектов сетевых приложений с графическим интерфейсом и реализации заданной функциональности. Готовый проект проходит тестирование и сдается преподавателю. Для прохождения аттестации в течение семестра студенты должны выполнить все обязательные упражнения.

Для оценки качества сданного проекта применяются следующие критерии оценивания

Оценка	Критерии
Отлично	<p>ОПК-3: Умеет спроектировать графический интерфейс и реализовать его с использованием подходящих элементов управления. Знает все рассмотренные на занятиях интерфейсы и протоколы сетевого программирования. Может использовать все варианты реализации механизма асинхронного обмена информацией. Программа выполняется для всех тестов.</p> <p>ПК-5: Может выбрать подходящие средства разработки и обосновать свой выбор. Свободно владеет справочной и дополнительной информацией из сети "Интернет".</p> <p>ПК-7: Создает полноценное сетевое приложение с графическим интерфейсом. Поясняет код и изменяет его при необходимости</p>
Хорошо	<p>ОПК-3: Умеет спроектировать графический интерфейс и реализовать его с использованием подходящих элементов управления. Знает основные интерфейсы и протоколы сетевого программирования. Может использовать несколько вариантов реализации механизма асинхронного обмена информацией. Программа выполняется для всех тестов, если какой-либо из тестов не выполняется программа дорабатывается.</p> <p>ПК-5: Может выбрать подходящие средства разработки. Владеет справочной и дополнительной информацией из сети "Интернет".</p> <p>ПК-7: Создает сетевое приложение с графическим интерфейсом.</p>

	Поясняет код и изменяет его при необходимости с небольшими неточностями.
Удовлетворительно	ОПК-3: В основном может спроектировать графический интерфейс и реализовать его с использованием подходящих элементов управления. Знает интерфейс Winsock и протоколы сетевого программирования TCP и UDP. Может использовать какой-либо вариант реализации механизма асинхронного обмена информацией. Программа выполняется для всех тестов, возможно кроме одного-двух крайних случаев. ПК-5: В основном может выбрать подходящие средства разработки. Умеет пользоваться справочной информацией. ПК-7: Создает сетевое приложение с графическим интерфейсом. С трудом поясняет код, не может изменить код при усложнении или существенном дополнении задачи
Неудовлетворительно	ОПК-3: Не может спроектировать графический интерфейс и реализовать его. Не знает интерфейса Winsock и протоколов сетевого программирования TCP и UDP. Не понимает механизма асинхронного обмена информацией. Программа не выполняется для большинства тестов. ПК-5: Не может выбрать подходящие средства разработки. Не умеет пользоваться справочной информацией. ПК-7: Не может создать сетевое приложение с графическим интерфейсом в среде разработки или создает неработающий проект. Не может пояснить код и изменить его.

## 1.2. Список вопросов по темам для проведения зачета в пятом семестре по курсу "Программирование в Windows и в сетях Windows"

### 1. Установка среды Visual Studio и справочной системы. Создание программ в среде Visual Studio

Установка Visual Studio и справочной системы MSDN.

Процесс компиляции и сборки программы. Создание проекта программы. Промежуточные файлы Visual Studio. Компоненты Visual Studio. Создание и редактирование исходных файлов программы. Изменение установок проекта. Построение программы. Выполнение программы. Отладка программы.

### 2. Модель программирования в Windows с использованием графического интерфейса

Обработка сообщений. Интерфейс графического устройства. Программирование, основанное на ресурсах. Управление памятью. Динамически подключаемые библиотеки. Интерфейс прикладных программ. Win32 API.

### 3. Создание программ с графическим интерфейсом

Генерация исходного кода. Редактирование исходного кода. Построение и запуск программы. Классы и файлы программы. Этапы выполнения программы. Исключения в библиотеке MFC.

### 4. Реализация представления

Реализация графического представления. Реализация текстового представления.

### 5. Реализация документа

Сохранение графических данных. Перерисовка окна. Добавление команд в меню. Удаление данных документа. Реализация команд меню.

## **6. *Хранение данных***

Принципы сохранения и загрузки данных документа из файлов на диске. Реализация стандартных команд меню File: New, Open, Save, Save As. Реализация технологии drag-and-drop.

## **7. *Прокрутка и разделение окон представления***

Добавление средств прокрутки окна. Добавление средств разделения окна. Обновление окна представления. Эффективная перерисовка.

## **8. *Перемещаемые панели и строки состояния***

Добавление в новую программу перемещаемой панели инструментов и строки состояния в процессе генерации кода. Добавление перемещаемой панели инструментов в существующую программу. Добавление строки состояния в существующую программу.

## **9. *Создание диалоговых окон***

Создание модальных диалоговых окон. Создание немодальных диалоговых окон. Создание диалоговых окон со вкладками. Диалоговые окна общего назначения.

## **10. *Разработка диалоговых приложений***

Простые диалоговые программы. Программы просмотра форм.

## **11. *Создание многодокументных приложений***

Многодокументный интерфейс. Создание MDI-программы в среде Developer Studio. Основные классы MDI-программы. Настройка ресурсов.

## **12. *Ввод/вывод символов***

Отображение текста. Чтение кодов символов, вводимых с клавиатуры. Управление курсором при редактировании.

## **13. *Использование функций рисования***

Создание объекта контекста устройства. Выбор средств рисования внутри объекта. Установка атрибутов рисования для объекта. Создание графических изображений. Функции рисования - члены класса CDC.

## **14. *Растровые изображения и битовые операции***

Создание растровых изображений. Выполнение битовых операций при отображении. Отображение значков.

## **15. *Печать и предварительный просмотр***

Добавление в программу средств печати и предварительного просмотра. Усовершенствованная печать.

## **16. *Многопоточные приложения***

Создание и управление вторичными потоками. Особенности использования MFC-классов в многопоточных программах. Синхронизация потоков.

## **17. *Связи между процессами***

Запуск новых процессов. Синхронизация процессов. Обмен данными по каналам. Совместное использование памяти. Использование буфера обмена.

### **1.3. Список вопросов по темам для проведения экзамена в шестом семестре по курсу "Программирование в Windows и в сетях Windows"**

#### **1. Механизм OLE**

Внедрение, связывание и автоматизация. Разработка программы-сервера. Разработка программы-контейнера.

#### **2. Создание и применение элементов ActiveX**

Разработка элементов ActiveX. Определение свойств, методов и событий элементов ActiveX. Разработка программы-контейнера элементов ActiveX.

#### **3. Динамически подключаемые библиотеки**

Основы DLL. Экспорт и импорт функций. DLL-расширения и обычные DLL.

#### **4. Интерфейс NetBIOS**

NetBIOS и сетевая модель OSI. Интерфейс Microsoft NetBIOS. Основы программирования NetBIOS. Типовые процедуры NetBIOS. Дейтаграммные операции. Дополнительные команды NetBIOS.

#### **5. Перенаправитель. Почтовые ящики**

Универсальные правила именования. Компоненты сетевого доступа. Протокол SMB. Имена почтовых ящиков. Использование архитектуры клиент-сервер. Дополнительные API-функции почтовых ящиков.

#### **6. Именованные каналы**

Правила именования каналов. Режимы передачи. Простой сервер и клиент. Усовершенствованный сервер каналов.

#### **7. Сетевые протоколы**

Интерфейс прикладного программирования Winsock. Характеристики протоколов. Сетевые протоколы, поддерживаемые Win32. Работа с Winsock.

#### **8. Семейства адресов и разрешение имен**

Протокол IP. Протоколы NetBIOS. Протоколы IPX/SPX. Инфракрасные сокет. Протокол ATM. Дополнительные функции Winsock 2.

#### **9. Основы интерфейса Winsock**

Инициализация Winsock. Проверка и обработка ошибок. Протоколы с установлением соединения: основные операции и соответствующие функции Win32 API. Протоколы, не требующие соединения.

#### **10. Ввод-вывод в Winsock**

Режимы работы сокетов. Модели управления вводом-выводом сокетов: select, WSAAsyncSelect, WSAEventSelect, модель перекрытого ввода-вывода, модель портов завершения.

#### **11. Параметры сокета и команды управления вводом-выводом**

Параметры сокета, их уровни. Функции ioctlsocket и WSAIocctl.

### **1.4. Правила проведения зачета в пятом семестре по курсу "Программирование в Windows и в сетях Windows"**

Для получения оценки "зачтено" студент должен выполнить следующие требования:

- в течение семестра выполнить и сдать все обязательные упражнения для самостоятельной работы;

- разработать и защитить собственный проект графического приложения, созданного в среде Microsoft Visual Studio с использованием библиотеки MFC, требования к содержанию проекта перечислены ниже;
- в процессе защиты своего проекта правильно ответить на вопросы по данному проекту, а также на вопросы для проведения зачета, перечисленные выше.

Основные требования к содержанию самостоятельного проекта для сдачи зачета

- Тема для приложения выбирается студентом самостоятельно, это может быть игрушка, специализированный редактор, приложение для какой-либо обработки данных, вычислений и т.п.
- Приложение должно быть создано в среде Microsoft Visual Studio с использованием библиотеки MFC, причем разработка ведется "с нуля", не допускается переделка ранее созданных кем-либо приложений. Приложение может иметь SDI, MDI или диалоговый интерфейс.
- Обязательные требования к устройству и функциональности приложения:
  - использование диалоговых окон для общения с пользователем;
  - связь переменных программы с элементами управления на основе механизма динамического обмена данными;
  - использование файлов и стандартных диалогов для сохранения и чтения данных в файлах;
- В качестве необязательных требований, повышающих оценку проекта можно указать использование своих динамически подгружаемых библиотек, многопоточности, развитой графики, возможно с анимацией и т.п.

Нередко игровые приложения, сданные в этом семестре, используются студентами в качестве основы клиентского приложения для сдачи самостоятельного проекта в следующем семестре.

В случае неудовлетворительной защиты проекта, либо его отсутствия, студент должен сдать зачет в специально назначенное для этого время. Ему потребуется в учебном классе в отведенный временной промежуток создать и отладить в среде Microsoft Visual Studio приложение со строго сформулированными характеристиками.

Оценка "зачтено" при этом выставляется только в случае удовлетворения всем требованиям, перечисленным в задании.

### **1.5. Правила проведения экзамена в шестом семестре по курсу "Программирование в Windows и в сетях Windows"**

Для получения положительной оценки студент должен выполнить следующие требования:

- в течение семестра выполнить и сдать все обязательные упражнения для самостоятельной работы;
- разработать и защитить собственный проект сетевого клиент-серверного приложения, созданного в среде Microsoft Visual Studio, требования к содержанию проекта перечислены ниже;
- в процессе защиты своего проекта правильно ответить на вопросы по данному проекту, а также на вопросы для проведения экзамена, перечисленные выше.

Основные требования к содержанию самостоятельного проекта для сдачи экзамена

- Тема для приложения выбирается студентом самостоятельно.
- Приложение должно иметь клиент-серверную архитектуру, основываться на интерфейсе Winsock и протоколах сетевого программирования TCP и/или UDP.

- Разработка серверной части ведется "с нуля", не допускается переделка ранее созданных кем-либо приложений. Хотя бы одна из частей (клиент или сервер) должна иметь графический интерфейс.
- Сервер должен обслуживать достаточно большое количество клиентов, можно ограничиться десятью.
- Обязательные требования к устройству и функциональности приложения:
  - использование в серверной части одной из асинхронных моделей ввода-вывода;
  - использование многопоточности с целью исключения блокирования пользовательского интерфейса;
  - корректное поведение любой части приложения в случае разрыва сетевой связи;
- Оценка за проект зависит от степени его развитости, качества интерфейса и полезности для пользователя.

В случае неудовлетворительной защиты проекта, либо его отсутствия, студент должен сдать экзамен в специально назначенное для этого время. Ему потребуется в учебном классе в отведенный временной промежуток отладить в среде Microsoft Visual Studio клиент-серверное приложение со строго сформулированными характеристиками. При этом студент получает работающие заготовки клиента и сервера из числа использованных в семестре учебных проектов. От студента требуется внести заданные изменения в сценарий работы приложения.

В случае сдачи экзамена в такой форме максимально возможная оценка "удовлетворительно" выставляется только в случае удовлетворения приложения всем требованиям, перечисленным в задании.

## **2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкалы оценивания**

### **2.1. Шкала оценивания сформированности компетенций и ее описание**

Оценивание уровня сформированности компетенций в процессе освоения дисциплины осуществляется по следующей трехуровневой шкале:

*Пороговый уровень* - предполагает отражение тех ожидаемых результатов, которые определяют минимальный набор знаний и (или) умений и (или) навыков, полученных студентом в результате освоения дисциплины. Пороговый уровень является обязательным уровнем для студента к моменту завершения им освоения данной дисциплины.

*Продвинутый уровень* - предполагает способность студента использовать знания, умения, навыки и (или) опыт деятельности, полученные при освоении дисциплины, для решения профессиональных задач. Продвинутый уровень превосходит пороговый уровень по нескольким существенным признакам.

*Высокий уровень* - предполагает способность студента использовать потенциал интегрированных знаний, умений, навыков и (или) опыта деятельности, полученных при освоении дисциплины, для творческого решения профессиональных задач и самостоятельного поиска новых подходов в их решении путем комбинирования и использования известных способов решения применительно к конкретным условиям. Высокий уровень превосходит пороговый уровень по всем существенным признакам.



## 2.2. Перечень компетенций, этапы их формирования, описание показателей и критериев оценивания компетенций на различных этапах их формирования

Код компетенции	Форма контроля	Этапы формирования (№ темы (раздела))	Показатели оценивания	Шкала и критерии оценивания компетенций на различных этапах их формирования		
				Пороговый уровень	Продвинутый уровень	Высокий уровень
<b>Общепрофессиональные компетенции</b>						
ОПК-3	Проверка выполнения учебных упражнений по темам № 2-27 Зачет Экзамен	2-27	<p><b>Знать:</b> – архитектуру основных типов Windows-приложений, понимать назначение основных классов, образующих эту архитектуру и уметь правильно их использовать для достижения требуемой функциональности; – язык программирования C++.</p> <p><b>Уметь:</b> – эффективно использовать основные классы библиотеки MFC для разработки Windows-приложений; – разрабатывать и отлаживать</p>	<p>1. Понимание модели выполнения в Windows.</p> <p>2. Знание принципов ООП и основных конструкций языка C++.</p> <p>3. Умение описать функциональные требования к системе, выполнить проектирование необходимых классов.</p> <p>4. Программирование требуемой функциональности классов.</p>	<p>1. Четкое понимание модели выполнения программ в Windows.</p> <p>2. Знание принципов ООП и конструкций языка C++. Понимание инкапсуляции, наследования, полиморфизма. Умение выбрать наиболее подходящие для конкретного случая средства языка.</p> <p>3. Умение описать функциональные требования к системе, выполнить проектирование необходимых классов, грамотно использовать модификаторы доступа.</p> <p>4. Программирование требуемой функциональности классов. Исключение</p>	<p>1. Четкое понимание модели выполнения программ в Windows.</p> <p>2. Знание принципов ООП и конструкций языка C++. Детальное понимание инкапсуляции, наследования, полиморфизма. Умение выбрать наиболее подходящие для конкретного случая средства языка. Четкое понимание достоинств и недостатков различных способов хранения и обработки данных класса.</p> <p>3. Умение описать функциональные требования к системе, выполнить проектирование необходимых классов, грамотно использовать модификаторы доступа.</p> <p>4. Программирование требуемой функциональности классов. Исключение дублирования</p>

		<p>многопоточные приложения, уметь использовать средства управления и механизмы синхронизации между потоками и процессами;</p> <p>– использовать для передачи информации по сети встроенные механизмы ОС Windows.</p> <p><b>Владеть:</b></p> <p>– навыками программирования, отладки и тестирования Windows-приложений в инструментальной среде Visual Studio.</p> <p>– навыками разработки сетевых приложений с использованием сетевых интерфейсов NETBIOS и Winsock;</p> <p>– навыками организации эффективного сетевого взаимодействия приложений с использова-</p>	<p>5. Умение проектировать пользовательский интерфейс.</p> <p>6. Понимание принципов и основных средств сетевого взаимодействия, навыки разработки сетевых приложений.</p>	<p>дублирования кода.</p> <p>5. Умение проектировать пользовательский интерфейс. Умение работать со всеми основными элементами управления.</p> <p>6. Понимание принципов и основных средств сетевого взаимодействия, навыки разработки сетевых приложений. Знание основных сетевых интерфейсов и протоколов. Понимание асинхронных моделей ввода-вывода.</p>	<p>кода. Умение создавать многопоточные приложения.</p> <p>5. Умение проектировать пользовательский интерфейс. Умение работать со всеми основными элементами управления. Умение организовать передачу данных между потоками и процессами.</p> <p>6. Понимание принципов и основных средств сетевого взаимодействия, навыки разработки сетевых приложений. Знание сетевых интерфейсов и протоколов. Умение организовать эффективное взаимодействие на базе асинхронных моделей ввода-вывода.</p>
--	--	--	--	--	---

			нием различных режимов работы сокетов и моделей ввода-вывода.			
<b>Общепрофессиональные компетенции</b>						
ПК-5	Проверка выполнения учебных упражнений по темам № 1-3, 24-27	1-3, 24-27	<p><b>Знать:</b> – современные языки программирования и средства разработки приложений для ОС Windows.</p> <p><b>Уметь:</b> – выбрать для решения конкретной задачи подходящий язык, среду разработки и библиотечные средства.</p> <p><b>Владеть навыками:</b> – поиска актуальной информации по языкам и средствам программирования в сети "Интернет".</p>	<p>1. Знание одного современного языка программирования и среды разработки приложений для ОС Windows. Умение использовать их для решения конкретной задачи.</p> <p>2. Владение базовыми навыками поиска информации по языкам и средствам программирования в сети "Интернет".</p>	<p>1. Знание нескольких современных языков программирования и средств разработки приложений для ОС Windows. Умение использовать их для решения конкретной задачи.</p> <p>2. Владение основными навыками поиска актуальной информации по языкам и средствам программирования в сети "Интернет".</p>	<p>1. Хорошее знание современных языков программирования и средств разработки приложений для ОС Windows. Умение выбрать для решения конкретной задачи подходящий язык, среду разработки и библиотечные средства.</p> <p>2. Уверенное владение навыками поиска актуальной информации по языкам и средствам программирования в сети "Интернет".</p>
ПК-7	Проверка выполнения учебных упражнений по темам № 2-27 Зачет Экзамен	2-27	<p><b>Знать:</b> – архитектуру основных типов Windows-приложений, понимать назначение основных классов, образующих эту архитектуру и уметь</p>	<p>1. Понимание модели выполнения в Windows.</p> <p>2. Знание принципов ООП и основных конструкций языка C++.</p>	<p>1. Четкое понимание модели выполнения программ в Windows.</p> <p>2. Знание принципов ООП и конструкций языка C++. Понимание инкапсуляции, наследования, полиморфизма. Умение выбрать наиболее подходящие для</p>	<p>1. Четкое понимание модели выполнения программ в Windows.</p> <p>2. Знание принципов ООП и конструкций языка C++. Детальное понимание инкапсуляции, наследования, полиморфизма. Умение выбрать наиболее подходящие</p>

		<p>правильно их использовать для достижения требуемой функциональности; – язык программирования C++.</p> <p><b>Уметь:</b> – эффективно использовать основные классы библиотеки MFC для разработки Windows-приложений; – разрабатывать и отлаживать многопоточные приложения, уметь использовать средства управления и механизмы синхронизации между потоками и процессами; – использовать для передачи информации по сети встроенные механизмы ОС Windows.</p> <p><b>Владеть:</b> – навыками программирования, отладки и тестирования Windows-</p>	<p>3. Умение описать функциональные требования к системе, выполнить проектирование необходимых классов.</p> <p>4. Программирование требуемой функциональности классов.</p> <p>5. Умение проектировать пользовательский интерфейс.</p> <p>6. Понимание принципов и основных средств сетевого взаимодействия, навыки разработки сетевых приложений.</p>	<p>конкретного случая средства языка.</p> <p>3. Умение описать функциональные требования к системе, выполнить проектирование необходимых классов, грамотно использовать модификаторы доступа.</p> <p>4. Программирование требуемой функциональности классов. Исключение дублирования кода.</p> <p>5. Умение проектировать пользовательский интерфейс. Умение работать со всеми основными элементами управления.</p> <p>6. Понимание принципов и основных средств сетевого взаимодействия, навыки разработки сетевых приложений. Знание основных сетевых интерфейсов и протоколов. Понимание асинхронных моделей ввода-вывода.</p>	<p>для конкретного случая средства языка. Четкое понимание достоинств и недостатков различных способов хранения и обработки данных класса.</p> <p>3. Умение описать функциональные требования к системе, выполнить проектирование необходимых классов, грамотно использовать модификаторы доступа.</p> <p>4. Программирование требуемой функциональности классов. Исключение дублирования кода. Умение создавать многопоточные приложения.</p> <p>5. Умение проектировать пользовательский интерфейс. Умение работать со всеми основными элементами управления. Умение организовать передачу данных между потоками и процессами.</p> <p>6. Понимание принципов и основных средств сетевого взаимодействия, навыки разработки сетевых приложений. Знание сетевых интерфейсов и протоколов. Умение организовать эффективное взаимодействие на базе асинхронных моделей</p>
--	--	--	---	---	--

			<p>приложений в инструментальной среде Visual Studio.</p> <ul style="list-style-type: none"><li>– навыками разработки сетевых приложений с использованием сетевых интерфейсов NETBIOS и Winsock;</li><li>– навыками организации эффективного сетевого взаимодействия приложений с использованием различных режимов работы сокетов и моделей ввода-вывода.</li></ul>			<p>ввода-вывода.</p>
--	--	--	---	--	--	----------------------

### **3. Методические рекомендации преподавателю по процедуре оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций**

Целью процедуры оценивания является определение степени овладения студентом ожидаемыми результатами обучения (знаниями, умениями, навыками и (или) опытом деятельности).

Процедура оценивания степени овладения студентом ожидаемыми результатами обучения осуществляется с помощью методических материалов, представленных в разделе «Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций»

#### **3.1 Критерии оценивания степени овладения знаниями, умениями, навыками и (или) опытом деятельности, определяющие уровни сформированности компетенций**

Пороговый уровень (общие характеристики):

- владение основным объемом знаний по программе дисциплины;
- знание основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы без существенных ошибок;
- владение инструментарием дисциплины, умение его использовать в решении стандартных (типовых) задач;
- способность самостоятельно применять типовые решения в рамках рабочей программы дисциплины;
- усвоение основной литературы, рекомендованной рабочей программой дисциплины;
- знание базовых теорий, концепций и направлений по изучаемой дисциплине;
- самостоятельная работа на практических и лабораторных занятиях, периодическое участие в групповых обсуждениях, достаточный уровень культуры исполнения заданий.

Продвинутый уровень (общие характеристики):

- достаточно полные и систематизированные знания в объеме программы дисциплины;
- использование основной терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать выводы;
- владение инструментарием дисциплины, умение его использовать в решении учебных и профессиональных задач;
- способность самостоятельно решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в базовых теориях, концепциях и направлениях по изучаемой дисциплине и давать им сравнительную оценку;
- самостоятельная работа на практических и лабораторных занятиях, участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

Высокий уровень (общие характеристики):

- систематизированные, глубокие и полные знания по всем разделам дисциплины;

- точное использование терминологии данной области знаний, стилистически грамотное, логически правильное изложение ответа на вопросы, умение делать обоснованные выводы;
- безупречное владение инструментарием дисциплины, умение его использовать в постановке и решении научных и профессиональных задач;
- способность самостоятельно и творчески решать сложные задачи (проблемы) в рамках рабочей программы дисциплины;
- полное и глубокое усвоение основной и дополнительной литературы, рекомендованной рабочей программой дисциплины;
- умение ориентироваться в основных теориях, концепциях и направлениях по изучаемой дисциплине и давать им критическую оценку;
- активная самостоятельная работа на практических и лабораторных занятиях, творческое участие в групповых обсуждениях, высокий уровень культуры исполнения заданий.

### **3.2 Описание процедуры выставления оценки**

В зависимости от уровня сформированности каждой компетенции по окончании освоения дисциплины студенту выставляется оценка. Для дисциплин, изучаемых в течение нескольких семестров, оценка может выставляться не только по окончании ее освоения, но и в промежуточных семестрах. Вид оценки («отлично», «хорошо», «удовлетворительно», «неудовлетворительно», «зачтено», «не зачтено») определяется рабочей программой дисциплины в соответствии с учебным планом.

Высокий уровень формирования компетенций соответствует оценке «отлично» за самостоятельные, контрольные работы и экзаменационную работу.

Продвинутый уровень формирования компетенций соответствует оценке «хорошо» за самостоятельные, контрольные работы и экзаменационную работу.

Пороговый уровень формирования компетенций соответствует оценке «удовлетворительно» за самостоятельные, контрольные работы и экзаменационную работу.

Оценка «отлично» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована на высоком уровне.

Оценка «хорошо» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на продвинутом уровне.

Оценка «удовлетворительно» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на пороговом уровне.

Оценка «неудовлетворительно» выставляется студенту, у которого хотя бы одна компетенция (полностью или частично формируемая данной дисциплиной) сформирована ниже, чем на пороговом уровне.

Оценка «зачет» выставляется студенту, у которого каждая компетенция (полностью или частично формируемая данной дисциплиной) сформирована не ниже, чем на пороговом уровне.

Оценка «незачтено» выставляется студенту, у которого хотя бы одна компетенция (полностью или частично формируемая данной дисциплиной) сформирована ниже, чем на пороговом уровне.

**Приложение №2**  
**к рабочей программе дисциплины**  
**"Программирование в Windows и в сетях Windows"**

**Методические указания для студентов по освоению дисциплины**

Основной формой изложения учебного материала по дисциплине "Программирование в Windows и в сетях Windows" являются лекции, проводимые в виде электронных презентаций, что позволяет сделать материал лекций более наглядными, улучшает информативность и понимание изучаемого курса.

По большинству тем предусмотрены учебные упражнения, на которых происходит закрепление лекционного материала путем применения его к конкретным задачам и получение навыков разработки и отладки Windows-приложений с использованием библиотеки MFC. Выполнение этих упражнений, а также выполнение индивидуальных заданий позволяет не только понять и закрепить теоретический материал, но и приобрести навык анализа предметной области и построения объектно-ориентированных Windows-приложений на языке C++.

Для успешного освоения дисциплины очень важно решение достаточно большого количества задач, как в аудитории, так и самостоятельно в качестве домашних заданий. Примеры решения задач разбираются и обсуждаются на лекциях и лабораторных занятиях. Основная цель решения задач – помочь усвоить фундаментальные понятия и основы концепции объектно-ориентированного анализа и проектирования. Для решения всех задач необходимо знать и понимать лекционный материал. Поэтому в процессе изучения дисциплины рекомендуется регулярное повторение пройденного лекционного материала. Материал, законспектированный на лекциях, необходимо дома еще раз прорабатывать и при необходимости дополнять информацией, полученной при выполнении лабораторных работ или из учебной литературы.

Большое внимание должно быть уделено выполнению домашней работы. В качестве заданий для самостоятельной работы дома студентам предлагаются задачи, аналогичные разобранным на лекциях и лабораторных занятиях или немного более сложные, которые являются результатом объединения нескольких базовых задач.

Для проверки и контроля усвоения теоретического материала, приобретенных практических навыков разработки Windows-приложений с использованием библиотеки MFC, в течение всего периода обучения проводятся консультации и разбор самостоятельного выполнения индивидуальных заданий.

В конце первого семестра студенты сдают зачет, в конце второго – экзамен. Зачет по итогам семестра выставляется по результатам выполнения учебных упражнений и индивидуальных заданий, а также краткого собеседования по вопросам теоретического материала. Экзамен предполагает самостоятельную разработку приложения и ответы на теоретические вопросы.

Освоить вопросы, излагаемые в процессе изучения дисциплины "Программирование в Windows и в сетях Windows" самостоятельно студенту достаточно сложно. Это связано и с существенно более высокой сложностью проектирования и отладки многопоточных приложений, необходимостью понимания средств и механизмов синхронизации. Поэтому посещение всех аудиторных занятий является совершенно необходимым. Без упорных и регулярных занятий в течение семестра сдать зачет и экзамен по итогам изучения дисциплины студенту будет сложно.



## Учебно-методическое обеспечение самостоятельной работы студентов по дисциплине

Для самостоятельной работы особенно рекомендуется использовать учебную литературу, указанную в разделе № 7 данной рабочей программы.

Также для подбора учебной литературы рекомендуется использовать широкий спектр интернет-ресурсов:

1. Электронно-библиотечная система «Университетская библиотека online» ([www.biblioclub.ru](http://www.biblioclub.ru)) - электронная библиотека, обеспечивающая доступ к наиболее востребованным материалам-первоисточникам, учебной, научной и художественной литературе ведущих издательств (\*регистрация в электронной библиотеке – только в сети университета. После регистрации работа с системой возможна с любой точки доступа в Internet.).

2. Для самостоятельного подбора литературы в библиотеке ЯрГУ рекомендуется использовать:

1. Личный кабинет ([http://lib.uniyar.ac.ru/opac/bk\\_login.php](http://lib.uniyar.ac.ru/opac/bk_login.php)) дает возможность получения on-line доступа к списку выданной в автоматизированном режиме литературы, просмотра и копирования электронных версий изданий сотрудников университета (учеб. и метод. пособия, тексты лекций и т.д.) Для работы в «Личном кабинете» необходимо зайти на сайт Научной библиотеки ЯрГУ с любой точки, имеющей доступ в Internet, в пункт меню «Электронный каталог»; пройти процедуру авторизации, выбрав вкладку «Авторизация», и заполнить представленные поля информации.

2. Электронная библиотека учебных материалов ЯрГУ ([http://www.lib.uniyar.ac.ru/opac/bk\\_cat\\_find.php](http://www.lib.uniyar.ac.ru/opac/bk_cat_find.php)) содержит более 2500 полных текстов учебных и учебно-методических материалов по основным изучаемым дисциплинам, изданных в университете. Доступ в сети университета, либо по логину/паролю.

3. Электронная картотека [«Книгообеспеченность»](http://www.lib.uniyar.ac.ru/opac/bk_bookreq_find.php) ([http://www.lib.uniyar.ac.ru/opac/bk\\_bookreq\\_find.php](http://www.lib.uniyar.ac.ru/opac/bk_bookreq_find.php)) раскрывает учебный фонд научной библиотеки ЯрГУ, предоставляет оперативную информацию о состоянии книгообеспеченности дисциплин основной и дополнительной литературой, а также цикла дисциплин и специальностей. Электронная картотека [«Книгообеспеченность»](http://www.lib.uniyar.ac.ru/opac/bk_bookreq_find.php) доступна в сети университета и через Личный кабинет.