

Документ подписан простой электронной подписью  
Информация о владельце:  
ФИО: Выборнова Лидия Александровна  
Должность: Ректор  
Дата подписания: 03.02.2022 15:17:47  
Уникальный программный ключ:  
c3b3b9c625f6c113afa2a2c42baff9e05a38b76e

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ  
УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ПОВОЛЖСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ СЕРВИСА»  
(ФГБОУ ВО «ПВГУС»)

Кафедра «Информационный и электронный сервис»

**РАБОЧАЯ УЧЕБНАЯ ПРОГРАММА**

по дисциплине  
«Программирование»

09.03.04 «Программная инженерия»  
направленности (профиля) «Разработка программно-информационных систем»

Тольятти 2018

Рабочая учебная программа по дисциплине «Программирование» включена в основную профессиональную образовательную программу направления подготовки 09.03.04 «Программная инженерия» направленности (профиля) «Разработка программно-информационных систем» решением Президиума Ученого совета

Протокол № 4 от 28.06.2018 г.

Начальник учебно-методического отдела \_\_\_\_\_  Н.М.Шемендюк  
28.06.2018 г.


Рабочая учебная программа по дисциплине «Программирование» разработана в соответствии с Федеральным государственным образовательным стандартом направления подготовки 09.03.04 «Программная инженерия» утвержденный приказом Министерства образования и науки Российской Федерации от 12 марта 2015 г. N 229

Составил: к.т.н., доцент Яницкая Т. С.

СОГЛАСОВАНО:


Директор научной библиотеки \_\_\_\_\_  В.Н.Еремина

СОГЛАСОВАНО:

Начальник управления информатизации \_\_\_\_\_  В.В.Обухов

Рабочая программа утверждена на заседании кафедры «Информационный и электронный сервис»

Протокол № 11 от «27» июня 2018 г.

Заведующий кафедрой \_\_\_\_\_  д.т.н., профессор В.И. Воловач  
(подпись)

СОГЛАСОВАНО:

Начальник учебно-методического отдела \_\_\_\_\_  Н.М.Шемендюк

# 1. Перечень планируемых результатов обучения по дисциплине (модулю, междисциплинарному курсу), соотнесенных с планируемыми результатами освоения образовательной программы

## 1.1. Цели освоения дисциплины

Целью освоения дисциплины является приобретение навыков в составлении программ на языках программирования, решающих разнообразные практические задачи, освоение основных методов современного программирования, изучение основ новых технологий в современном программировании.

1.2. В соответствии с видами профессиональной деятельности, на которые ориентирована образовательная программа указанного направления подготовки, содержание дисциплины позволит обучающимся решать следующие профессиональные задачи:

### **производственно-технологическая деятельность:**

- обеспечение соответствия разрабатываемого программного обеспечения и технической документации российским и международным стандартам, техническим условиям, ведомственным нормативным документам и стандартам предприятия;
- участие в процессе разработки программного обеспечения.

## 1.3. Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины у обучающихся формируются следующие компетенции:

Код компетенции	Наименование компетенции	Направление подготовки
ПК-1	Готовность применять основные методы и инструменты разработки программного обеспечения.	09.03.04 «Программная инженерия», направленность (профиль) «Разработка программно-информационных систем»
ПК-3	Владение навыками использования различных технологий разработки программного обеспечения.	09.03.04 «Программная инженерия», направленность (профиль) «Разработка программно-информационных систем»

## 1.4. Перечень планируемых результатов обучения по дисциплине

Результаты освоения дисциплины	Технологии формирования компетенции по указанным результатам	Средства и технологии оценки по указанным результатам
<b>Знает:</b> Основы и методы решения задач объектно-ориентированного	Лекции	Собеседование

программирования; (ПК-1) основы разработки программного кода с использованием объектно-ориентированных языков программирования (ПК-3)		
<b>Умеет:</b> Использовать средства реализации информационных технологий (методические, информационные, математические, алгоритмические, технические и программные) (ПК-1) Применять различные объектно-ориентированные языки программирования для разработки программного обеспечения (ПК-3)	Лабораторные работы	Собеседование Защита лабораторных работ
<b>Имеет практический опыт:</b> Решения основных задач объектно-ориентированного программирования (ПК-1) Разработки программного обеспечения с использованием различных объектно-ориентированных языков программирования (ПК-3)	Лекции Лабораторные работы	Защита лабораторных работ

## 2. Место дисциплины в структуре образовательной программы

Дисциплина относится к вариативной части.

Ее освоение осуществляется в 3 и 4 семестрах (очная и заочная форма); в 4 и 5 семестры (заочная форма (февраль)).

№ п/п	Наименование дисциплин, определяющих междисциплинарные связи	Код компетенции(й)
	Предшествующие дисциплины	
1	Специальные разделы информатики	ОПК-1, ОПК-3
	Последующие дисциплины:	
1	Командный проект по программной инженерии	ПК-1, ПК-2, ПК-3, ПК-6, ПК-7, ПК-9

## 3. Объем дисциплины в зачетных единицах с указанием количества академических часов, выделенных на контактную работу обучающихся с преподавателем (по видам учебных занятий) и на самостоятельную работу

Распределение фонда времени по семестрам и видам занятий

Виды занятий	очная форма обучения	заочная форма обучения (февраль)	заочная форма обучения
Итого часов	252 ч.	252 ч.	252 ч.
Зачетных единиц	7 з.е.	7 з.е.	7 з.е.
Лекции (час)	16/24	4/6	4/6
Практические (семинарские) занятия (час)	-	-	-

Лабораторные работы (час)	30/32	10/6	10/6
Самостоятельная работа (час)	98/27	126/87	126/87
Курсовой проект (работа) (+,-)	КР (4 сем.)	КР (5 сем.)	КР (4 сем.)
Контрольная работа (+,-)	-	-	-
Экзамен, семестр /час.	4/27	5/9	4/9
Зачет, семестр	3	4 сем./3 ч.	3 сем./3 ч.
Контрольная работа, семестр	-	-	-

#### 4. Содержание дисциплины, структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

##### 4.1. Содержание дисциплины

№ п/п	Раздел дисциплины	Виды учебных занятий, включая самостоятельную работу студентов и трудоемкость (в академических часах)					Средства и технологии оценки
		Лекции, час	Практические (семинарские) занятия, час	Лабораторные работы, час	Самостоятельная работа, час		
<b>3/4/3 семестр</b>							
1	Структурные, объектные, логические языки программирования. Принципы объектно-ориентированного подхода. Знакомство со средой разработки приложений NetBeans	2/0/0	-	-	0/3/3	Конспект	
2	Лексика языка Java. Операции. Литералы. Типы данных. Операторы. Работа с примитивными переменными	1/1/1	-	2/2/2	0/3/3	Отчет по лабораторным работам	
3	Работа с объектами	1/0/0	-	-	5/15/15	Конспект	
4	Использование операторов и управляющих конструкций	1/1/1	-	2/2/2	5/15/15	Отчет по лабораторным работам	
5	Создание и использование массивов	1/1/1	-	4/4/4	10/10/10	Отчет по лабораторным работам	
6	Использование циклов	1/1/1	-	2/2/2	10/10/10	Отчет по лабораторным работам	
7	Использование инкапсуляции и конструкторов	1/0/0	-	4/0/0	10/10/10	Отчет по лабораторным работам	
8	Обработка ошибок	1/0/0	-	-	10/10/10	Отчет по лабораторным работам	
9	Объявление классов	1/0/0	-	4/0/0	10/10/10	Отчет по лабораторным работам	
10	Инкапсуляция и полиморфизм	2/0/0	-	4/0/0	10/10/10	Отчет по	

						лабораторным работам
11	Проектирование Java классов	2/0/0	-	4/0/0	10/10/10	Отчет по лабораторным работам
12	Дополнительные вопросы проектирования классов	2/0/0	-	4/0/0	18/20/20	Отчет по лабораторным работам
	<b>Промежуточная аттестация по дисциплине за 3/4/3 семестр</b>	<b>16/4/4</b>	<b>-</b>	<b>30/10/10</b>	<b>98/126/126</b>	<b>Зачет</b>
<b>4/5/4 семестр</b>						
13	Наследование с использованием Java интерфейсов	2/0/0	-	4/0/0	2/3/3	Отчет по лабораторным работам
14	Настраиваемые типы и коллекции	2/0/0	-	4/0/0	2/4/4	Отчет по лабораторным работам
15	Обработка строк	2/1/1	-	4/0/0	2/10/10	Отчет по лабораторным работам
16	Исключения и Assertions	2/1/1	-	4/0/0	2/10/10	Отчет по лабораторным работам
17	Основы ввода/вывода	2/1/1	-	4/2/2	2/10/10	Отчет по лабораторным работам
18	Файловый ввод/вывод с использованием NIO 2	4/0/0	-	4/0/0	2/10/10	Отчет по лабораторным работам
19	Многопоточность	2/1/1	-	4/2/2	2/10/10	Отчет по лабораторным работам
20	Конкурентность (параллелизм)	2/1/1	-	-	3/10/10	Отчет по лабораторным работам
21	Приложения базы данных с использованием JDBC	4/1/1	-	4/2/2	4/10/10	Отчет по лабораторным работам
22	Локализация	2/0/0	-	-	4/10/10	Отчет по лабораторным работам
	<b>Промежуточная аттестация по дисциплине за 4/5/4 семестр</b>	<b>24/6/6</b>	<b>-</b>	<b>32/6/6</b>	<b>27/87/87</b>	<b>Экзамен</b>
	<b>Итого</b>	<b>40/10/10</b>	<b>-</b>	<b>62/16/16</b>	<b>123/213/213</b>	

Примечание:

-/-/-, объем часов соответственно для очной формы обучения, заочной формы обучения (февраль), заочной формы обучения

#### 4.2.Содержание практических (семинарских) занятий

Практические работы планом не предусмотрены.

### 4.3.Содержание лабораторных работ

№	Наименование лабораторных работ	Объем часов	Наименование темы дисциплины
<b>3/4/3 семестр</b>			
1	Лабораторная работа 1. Создание простейшего консольного приложения.	2/2/2	Лексика языка Java. Операции. Литералы. Типы данных. Операторы. Работа с примитивными переменными.
2	Лабораторная работа 2. Создание приложения с использованием управляющих конструкций.	2/2/2	Использование операторов и управляющих конструкций.
3	Лабораторная работа 3. Создание приложения для работы с массивами.	4/4/4	Создание и использование массивов.
4	Лабораторная работа 4. Создание приложения для работы с циклами.	2/2/2	Использование циклов.
5	Лабораторная работа 5. Создание конструкторов. Реализация инкопсуляции.	4/0/0	Использование инкапсуляции и конструкторов.
6	Лабораторная работа 6. Создание простого Java-класса.	4/0/0	Объявление классов.
7	Лабораторная работа 7. Проектирование Java-классов. Создание подклассов.	4/0/0	Инкапсуляция и полиморфизм.
8	Лабораторная работа 8. Создание Java-классов.	4/0/0	Проектирование Java классов.
9	Лабораторная работа 9. Использование шаблона проектирования Singleton. Проектирование абстрактных и вложенных классов.	4/0/0	Дополнительные вопросы проектирования классов.
<b>Итого за 3/4/3 семестр</b>		30/10/10	
<b>4/5/4 семестр</b>			
1	Лабораторная работа 1. Реализация интерфейса. Применение паттерна DAO.	4/0/0	Наследование с использованием Java интерфейсов.
2	Лабораторная работа 2. Подсчет номера деталей с помощью HashMap.	4/0/0	Настраиваемые типы и коллекции.
3	Лабораторная работа 3. Разбор текста с помощью split().	4/0/0	Обработка строк.
4	Лабораторная работа 4. Обработка исключений.	4/0/0	Исключения и Assertions.
5	Лабораторная работа 5. Написание простой консоли I/O приложения.	4/2/2	Основы ввода/вывода.
6	Лабораторная работа 6. Написание слияние файлов приложения. Рекурсивное копирование. Использование PathMatcher для рекурсивного удаления.	4/0/0	Файловый ввод/вывод с использованием NIO 2.
7	Лабораторная работа 7. Синхронизация доступа к разделяемым данным. Реализация многопоточной программы.	4/2/2	Многопоточность.
8	Лабораторная работа 8. Построение базы данных приложения с помощью JDBC.	4/2/2	Приложения базы данных с использованием JDBC.
<b>Итого за 4/5/4 семестр</b>		32/6/6	
<b>Итого</b>		62/16/16	



–/–/–, объем часов соответственно для очной формы обучения, заочной формы обучения (февраль), заочной формы обучения

## 5. Учебно-методическое обеспечение самостоятельной работы обучающихся по дисциплине

### Технологическая карта самостоятельной работы студента

Код реализуемой компетенции	Вид деятельности студентов (задания на самостоятельную работу)	Итоговый продукт самостоятельной работы	Средства и технологии оценки	Объем часов
1	2	3	4	5
ПК-1, ПК-3	Выполнение индивидуальных заданий в виде реферата, презентации и доклада на заданную тему.	Реферат, презентация, доклад	Собеседование	98/126/126
<b>Итого за 3/4/3 семестр</b>				98/126/126
ПК-1, ПК-3	Выполнение индивидуальных заданий в виде реферата, презентации и доклада на заданную тему.	Реферат, презентация, доклад	Собеседование	25/87/87
<b>Итого за 4/5/4 семестр</b>				25/87/87
<b>Итого</b>				123/213/213

Примечание:

–/–/–, объем часов соответственно для очной формы обучения, заочной формы обучения (февраль), заочной формы обучения

#### Рекомендуемая литература:

1. Блох, Д. Java. Эффективное программирование [Текст] / Д. Блох ; пер. Е. Коротылев. - 2-е изд. - М. : Лори. - 2015. - 294 с.
2. Васильев, А. Н. Java. Объектно-ориентированное программирование [Текст] : базовый курс по объект.-ориентир. прогр. для магистров и бакалавров / А. Н. Васильев. - СПб. : Питер. - 2014. - 396 с.
3. Программирование[Электронный ресурс] : учеб. для студентов вузов по направлению "Пед. образование" : в 2 т. Т. 1 / Э. А. Нигматулина [и др.] ; под ред. Н. И. Пака. - Документ AdobeAcrobat. - М. : Академия, 2013. - 63,9 МБ, 267 с. - Режим доступа: <http://elib.tolgas.ru>.
4. Программирование[Электронный ресурс] : учеб. для студентов вузов по направлению "Пед. образование" : в 2 т. Т. 2 / Э. А. Нигматулина [и др.] ; под ред. Н. И. Пака. - Документ AdobeAcrobat. - М. : Академия, 2013. - 60,4 МБ, 240 с. : ил. - Режим доступа: <http://elib.tolgas.ru>.
5. Фейерштейн, С. Oracle PL/SQL для профессионалов [Текст] / С. Фейерштейн, Б. Прибыл ; [пер. с англ. Е. Матвеев]. - 6-е изд. - СПб. : Питер. - 2015. - 1024 с.

#### Содержание заданий для самостоятельной работы

### 3 семестр

1. Изучить объектную модель Java.
2. Изучить управляющие конструкции в языке Java.
3. Изучить принципы работы с массивами в Java.
4. Изучить возможности работы с циклами в Java.
5. Изучить работу с конструкторами в Java.
6. Изучить виды ошибок, возможности Java для обработки исключений.
7. Изучить построение классов в Java.
8. Рассмотреть свойства инкапсуляции и полиморфизма в Java классах.
9. Рассмотреть основы создания Java классов.
10. Рассмотреть вопросы создания Java классов.

### 4 семестр

1. Изучить вопросы наследования с использованием Java интерфейсов.
2. Изучить возможности работы с настраиваемыми типами данных в Java.
3. Изучить возможности работы со строками в Java.
4. Изучить вопросы обработки исключений в Java.
5. Изучить основы ввода/вывода в Java.
6. Изучить основы файлового ввода/вывода с использованием NIO 2.
7. Изучить возможности разработки многопоточковых приложений.
8. Рассмотреть возможности работы с потоковыми приложениями с использованием механизма конкурентности.
9. Изучить возможности работы с базами данных в Java с использованием JDBC.
10. Изучить основы локализации, преимуществ.

### 6. Методические указания для обучающихся по освоению дисциплины Инновационные образовательные технологии

Вид образовательных технологий, средств передачи знаний, формирования умений и практического опыта	№ темы / тема лекции	№ практического (семинарского) занятия/наименование темы	№ лабораторной работы / цель
Разбор конкретных ситуаций	-	-	1-17
Слайд-лекции	1-22	-	-

В начале семестра студентам необходимо ознакомиться с технологической картой дисциплины, выяснить, какие результаты освоения дисциплины заявлены (знания, умения, практический опыт). Для успешного освоения дисциплины студентам необходимо выполнить задания, предусмотренные рабочей учебной программой дисциплины и пройти контрольные точки в сроки, указанные в технологической карте (раздел 11). От качества и полноты их выполнения будет зависеть уровень сформированности компетенции и оценка текущей успеваемости по дисциплине. По итогам текущей успеваемости студенту может быть выставлена оценка по промежуточной аттестации, если это предусмотрено технологической картой дисциплины. Списки учебных пособий, научных трудов, которые студентам следует прочесть и законспектировать, темы лабораторных работ и вопросы к ним, вопросы к экзамену (зачету) и другие необходимые материалы указаны в разработанном для данной дисциплины учебно-методическом комплексе.

Основной формой освоения дисциплины является контактная работа с преподавателем - лекции, лабораторные работы, консультации (в том числе индивидуальные), в том числе проводимые с применением дистанционных технологий.

По дисциплине часть тем (разделов) изучается студентами самостоятельно. Самостоятельная работа предусматривает подготовку к аудиторным занятиям, выполнение заданий (письменных работ, творческих проектов и др.) подготовку к промежуточной аттестации (экзамену (зачету)).

На лекционных занятиях и лабораторных работах вырабатываются навыки и умения обучающихся по применению полученных знаний в конкретных ситуациях, связанных с будущей профессиональной деятельностью. По окончании изучения дисциплины проводится промежуточная аттестация (экзамен, (зачет)).

Регулярное посещение аудиторных занятий не только способствует успешному овладению знаниями, но и помогает организовать время, т.к. все виды учебных занятий распределены в семестре планомерно, с учетом необходимых временных затрат.

### **6.1. Методические указания для обучающихся по освоению дисциплины на лабораторных работах**

Практические занятия планом не предусмотрены.

#### **Лабораторные работы**

№	Наименование лабораторных работ	Задание по лабораторным работам
<b>3 семестр</b>		
1	Лабораторная работа 1. Создание простейшего консольного приложения.	Объявление и инициализация переменных полей. Переменные типы данных, такие как целые, с плавающей точкой, текстовые и логические. Объявление переменных и присваивание значений. Использование констант. Использование арифметических операторов для модификации значений.
2	Лабораторная работа 2. Создание приложения с использованием управляющих конструкций.	Использование операторов отношения и условных операторов. Проверка равенства строк. Определение различных условий в программе и определение алгоритма. Создание конструкций if и if/else. Вложение и создание цепочек условных предложений. Использование предложения switch.
3	Лабораторная работа 3. Создания приложения для работы с массивами.	Объявление, создание и инициализация одномерного массива. Объявление, создание и инициализация двумерного массива. Использование цикла for для обработки массива. Создание и инициализация ArrayList. Использование предложения import для работы с существующими Java API. Доступ к значению в массиве или ArrayList. Использование массива аргументов.
4	Лабораторная работа 4. Создание приложения для работы с циклами.	Создание циклов while и вложенных циклов while. Создание цикла for. Использование ArrayLists с циклами for. Создание цикла do while. Понятие об области действия переменной
5	Лабораторная работа 5. Создание	Создание конструкторов. Реализация

	конструктов. Реализация инкапсуляции.	инкапсуляции.
6	Лабораторная работа 6. Создание простого Java-класса.	Простые Java классы. Java поля, конструкторы и методы. Моделирование объектов с помощью Java классов. Предложения package и import.
7	Лабораторная работа 7. Проектирование Java-классов. Создание подклассов.	Инкапсуляция при проектировании Java класса. Моделирование задач бизнеса с помощью Java классов. Неизменяемость. Создание подклассов. Перегрузка методов. Методы с переменными аргументами.
8	Лабораторная работа 8. Создание Java-классов.	Модификаторы доступа: private, protected и public. Переопределение метода. Перегрузка конструктора. Оператор instanceof. Вызов виртуального метода. Полиморфизм. Приведение объектных ссылок. Переопределение методов Object.
9	Лабораторная работа 9. Использование шаблона проектирования Singleton. Проектирование абстрактных и вложенных классов.	Абстрактные классы и обобщение типов. Модификаторы static и final. Модификаторы полей - общие рекомендации. Шаблон проектирования Singleton. Проектирование абстрактных классов. Вложенные классы. Перечислимые типы.
<b>4 семестр</b>		
10	Лабораторная работа 1. Реализация интерфейса. Применение паттерна DAO.	Java интерфейсы. Типы наследования. Композиция объектов и делегирование метода. Реализация нескольких интерфейсов. Шаблон проектирования DAO.
11	Лабораторная работа 2. Подсчет номера деталей с помощью HashMap.	Настраиваемые классы и параметры типов. Вывод типа (diamond). Коллекции и настраиваемые типы List, set и Map Stack и Deque.
12	Лабораторная работа 3. Разбор текста с помощью split().	Манипулирование строками с помощью StringBuilder и StringBuffer. Основные методы String. Разбор текста в Java. Обработка ввода с помощью Scanner. Вывод текста и форматирование. Регулярные выражения с использованием классов Pattern и Matcher.
13	Лабораторная работа 4. Обработка исключений.	Категории исключений. Классы стандартных исключений Java. Создание пользовательских классов исключений. Использование предложений try-catch и finally. Использование try-with-resources и интерфейса AutoCloseable. Возможность multi-catch. Обработка исключений - общие рекомендации Assertions.
14	Лабораторная работа 5. Написание простой консоли I/O приложения.	I/O средствами Java. Чтение входного потока консоли. Запись на console. Использование I/O потоков. Цепочки I/O потоков. Канальный I/O. Запись и чтение объектов с использованием сериализации.
15	Лабораторная работа 6. Написание слияние	Интерфейс Path. Класс Files. Операции

	файлов приложения. Рекурсивное копирование. Использование PathMatcher для рекурсивного удаления.	Directory и File. Управление атрибутами файловой системы. Чтение, запись и создание файлов. Отслеживание изменений файловой системы for file system changes.
16	Лабораторная работа 7. Синхронизация доступа к разделяемым данным. Реализация многопоточной программы.	Планирование выполнения заданий операционной системы. Многопоточное окружение. Создание многопоточковых решений. Общий доступ потоков к данным. Синхронизация и взаимная блокировка. Неизменяемые объекты.
17	Лабораторная работа 8. Построение базы данных приложения с помощью JDBC.	Макет JDBC API. JDBC драйверы. Запросы и результаты. PreparedStatement и CallableStatement. Транзакции RowSet 1.1 RowSetProvider и RowSetFactory. Шаблон DAO и JDBC.

Лабораторные работы обеспечивают:

формирование умений и навыков обращения с приборами и другим оборудованием, демонстрацию применения теоретических знаний на практике, закрепление и углубление теоретических знаний, контроль знаний и умений в формулировании выводов, развитие интереса к изучаемой дисциплине.

Применение лабораторных работ позволяет вовлечь в активную работу всех обучающихся группы и сформировать интерес к изучению дисциплины.

Самостоятельный поиск ответов на поставленные вопросы и задачи в ходе лабораторной работы приобретают особую значимость в восприятии, понимании содержания дисциплины.

Изученный на лекциях материал лучше усваивается, лабораторные работы демонстрируют практическое их применение.

## **6.2. Методические указания для выполнения контрольных работ (письменных работ)**

Контрольные работы учебным планом не предусмотрены.

## **6.3. Методические указания для выполнения курсовых работ**

Курсовая работа, рассматриваются как вид учебной работы по дисциплине и выполняются в пределах часов, отводимых на ее изучение. Выполнение курсовых работ по дисциплинам осуществляется в соответствии с тематикой, сформированной в соответствии с содержанием дисциплины, сопряженным с направленностью (профилем) образовательной программы. Подготовка курсовой работы содействует лучшему усвоению обучающимися учебного материала, формирует практический опыт и умения по изучаемой дисциплине, способствует формированию у обучающихся навыков поиска и критического анализа научной литературы, готовит их к самостоятельной профессиональной деятельности, повышает уровень профессиональной подготовки, является подготовительным этапом к написанию выпускником выпускной квалификационной работы.

Выполнение курсовых работ предусматривается по дисциплинам, формирующим последовательно профессиональные компетенции выпускника, и служит основой для выполнения выпускной квалификационной работы.

Курсовая работа выполняется индивидуально каждым студентом. В результате выполнения курсовой работы должно быть разработано программное обеспечение (в виде исходных кодов) и оформлена пояснительная записка (35~35 стр.) (в печатном и электронном (.doc/.docx) виде).

1. Пояснительная записка должна содержать следующие разделы:

- титульный лист;

- лист задания, лист рецензии
  - введение — постановка задачи;
  - теоретический раздел — описание выбранной предметной области, средствах разработки, общие сведения о (реляционных) БД;
  - практический раздел — алгоритмы работы программы; описание операций, выполняемых программой, объекты ООП, методы, свойства.
  - выводы — краткое резюме по проделанной работе; преимущества и недостатки.
  - список литературы.
2. Исходный код программного обеспечения должен:
    - разрабатываться на языке Java;
    - содержать корректную (полноценную) обработку ошибок;
    - выполнять все поставленные задачи.
  3. Требования к базе данных
    - база данных должна содержать не менее 3 связанных таблиц.

Выбор темы курсового проекта: за учащимся закрепляется тема на основе его пожеланий и обсуждения с преподавателем. При возникновении у обучающегося затруднений с выбором темы, подбором литературы, составлением плана проекта необходимую помощь в этих вопросах ему оказывает преподаватель, являющийся руководителем выполнения курсового проекта.

Выбранная тема курсового проекта фиксируется на кафедре в приложении к распоряжению декана факультета. Тема курсового проекта не должна повторяться в одной учебной группе. Для студентов заочной формы обучения выбор темы курсового проекта производится из утвержденной тематики в соответствии с одним или двумя последними номерами зачетной книжки.

Библиографический список в работе помещается после заключения. Каждый источник, упомянутый в списке, значится под определенным порядковым номером и должен быть описан в соответствии с ГОСТом 7.1-2003. («Библиографическое описание документа»). При оформлении курсового проекта (работы), используется алфавитный способ группировки библиографических описаний.

Разработать приложение на языке Java в среде программирования NetBeans.

Курсовая работа предоставляется в электронном и распечатанном виде. Также предоставляются в электронном виде следующие файлы:

- Разработанное приложение среде программирования NetBeans
- Презентация 10 слайдов
- Доклад к презентации на 10-15 минут

### **Примерная тематика курсовой работы**

1. Анимация алгоритма устойчивой сортировки. Сортировка выбором (англ. Selection sort) — поиск наименьшего или наибольшего элемента и помещение его в начало или конец упорядоченного списка.
2. Анимация алгоритма устойчивой сортировки. Сортировка пузырьком (англ. Bubblesort) — для каждой пары индексов производится обмен, если элементы расположены не по порядку.
3. Анимация алгоритма устойчивой сортировки. Сортировка перемешиванием (англ. Cocktailsort).
4. Анимация алгоритма устойчивой сортировки. Гномья сортировка (англ. Gnomesort). — схожа с сортировкой пузырьком и сортировкой вставками.
5. Анимация алгоритма устойчивой сортировки. Сортировка вставками (Insertionsort) — Определяем, где текущий элемент должен находиться в упорядоченном списке, и вставляем его туда.
6. Анимация алгоритма устойчивой сортировки. Сортировка слиянием (Mergesort) — выстраиваем первую и вторую половину списка отдельно, а затем объединяем упорядоченные списки.

7. Анимация алгоритма устойчивой сортировки. Сортировка с помощью двоичного дерева (англ. Treесort).
8. Анимация алгоритма устойчивой сортировки. Блочная сортировка (Корзинная сортировка, Bucketsort)
9. Анимация алгоритма устойчивой сортировки. Сортировка подсчётом (Countingsort).
10. Анимация алгоритма неустойчивой сортировки. Сортировка Шелла (Shellsort).
11. Анимация алгоритма устойчивой сортировки. Сортировка расчёской (Combsort)
12. Анимация алгоритма устойчивой сортировки. Пирамидальная сортировка (сортировка кучи, Heapsort) превращаем список в кучу, берём наибольший элемент и добавляем его в конец списка
13. Анимация алгоритма устойчивой сортировки. Плавная сортировка (Smoothsort)
14. Анимация алгоритма устойчивой сортировки. Быстрая сортировка (Quicksort), в варианте с минимальными затратами памяти.
15. Анимация алгоритма устойчивой сортировки. Интроспективная сортировка (Introsort)
16. Анимация алгоритма устойчивой сортировки. Терпеливая сортировка (Patience sorting)
17. Анимация алгоритма устойчивой сортировки. Поразрядная сортировка (она же цифровая сортировка).
18. Анимация алгоритма непрактичной сортировки. Bogosort. Произвольно перемешать массив, проверить порядок.
19. Анимация алгоритма непрактичной сортировки. Сортировка перестановкой. Для каждой пары осуществляется проверка верного порядка и генерируются всевозможные перестановки исходного массива.
20. Анимация алгоритма непрактичной сортировки. Глупая сортировка (Stupidsort) .

## 7. Фонд оценочных средств для проведения текущего контроля успеваемости и промежуточной аттестации обучающихся по дисциплине (зачёт, экзамен)

Фонды оценочных средств, позволяющие оценить уровень сформированности компетенций и результаты освоения дисциплины, представлены следующими компонентами:

Код оцениваемой компетенции и (или ее части)	Тип контроля	Вид контроля	Количество Элементов, шт.
ПК-1, ПК-3	текущий	устный опрос	1-7
ПК-1, ПК-3	промежуточный	тест	1-80

### 7.1. Оценочные средства для текущего контроля успеваемости, промежуточной аттестации по итогам освоения дисциплины

Результаты освоения дисциплины	Оценочные средства (перечень вопросов, заданий и др.)
<b>Знает:</b> Основы и методы решения задач объектно-ориентированного программирования; (ПК-1)	1. Особенности языка и платформы Java. 2. Классификация программ по типу исполнения (компилируемые, интерпретируемые, исполняемые на виртуальных машинах).

<p>основы разработки программного кода с использованием объектно-ориентированных языков программирования (ПК-3)</p>	<p>Виртуальная машина Java. JIT-компиляция.</p> <ol style="list-style-type: none"> <li>3. Создание простейшей программы на Java, её компиляция в байт-код и запуск.</li> <li>4. Средства разработки Java-приложений. Интегрированные среды разработки.</li> <li>5. Встроенные типы данных. Способы задания литералов различных типов.</li> <li>6. Хранение данных в памяти ЭВМ.</li> <li>7. Приведение типов (явное и автоматическое). Константы и переменные.</li> <li>8. Оператор присваивания. Порядок действий (приоритет операторов).</li> <li>9. Арифметические операторы. Операторы инкремента и декремента.</li> <li>10. Операторы сравнения и логические операторы.</li> <li>11. Операторы ветвления. Условный оператор. Минимизация количества проверок.</li> <li>12. Встроенный класс String. Строковые операции.</li> <li>13. Ввод, вывод данных.</li> <li>14. Операторы организации циклов. Цикл типа «n раз».</li> <li>15. Операторы организации циклов. Цикл типа «пока» (с пред- и постпроверкой условия).</li> <li>16. Массивы. Способы объявления и инициализации массивов. Индексация и размер массива.</li> <li>17. Массивы. Многомерные массивы.</li> <li>18. Процедурное программирование. Объектно-ориентированное программирование.</li> <li>19. Основные понятия ООП. Объекты и классы. Абстракция данных. Сценарий построения объектно-ориентированной программы.</li> <li>20. Классы. Члены классов. Методы и поля.</li> <li>21. Специальные методы классов (конструкторы). Конструктор по умолчанию.</li> <li>22. Модификаторы уровня доступа (default, public, protected, private).</li> <li>23. основополагающие принципы ООП. Инкапсуляция.</li> <li>24. основополагающие принципы ООП. Наследование. Управление наследованием.</li> <li>25. Интерфейсы как средство реализации множественного наследования.</li> <li>26. основополагающие принципы ООП. Полиморфизм. Средства реализации полиморфизма.</li> <li>27. Иерархия классов Java. Коренной класс Object и его методы.</li> </ol>
<p><b>Умеет:</b> Использовать средства реализации информационных технологий (методические, информационные, математические, алгоритмические, технические и программные) (ПК-1) Применять различные объектно-ориентированные языки</p>	<ol style="list-style-type: none"> <li>1. Встроенный класс String. Строковые операции.</li> <li>2. Стандартные потоки ввода-вывода. Организация ввода и вывода данных. Класс Scanner.</li> <li>3. Операторы организации циклов. Цикл типа «n раз».</li> <li>4. Операторы организации циклов. Цикл типа «пока» (с пред- и постпроверкой условия).</li> <li>5. Массивы. Способы объявления и инициализации массивов. Индексация и размер массива.</li> </ol>



программирования для разработки программного обеспечения (ПК-3)

6. Массивы. Алгоритмы сортировки.
7. Массивы. Многомерные массивы.
8. Статические методы классов. Методы функционального и процедурного типа.
9. Сигнатура метода. Перегрузка методов.
10. Процедурное программирование. Объектно ориентированное программирование. Сравнение парадигм.
11. Основные понятия ООП. Объекты и классы. Абстракция данных. Сценарий построения объектно-ориентированной программы.
12. Члены классов. Методы и поля.
13. Специальные методы классов (конструкторы). Конструктор по умолчанию.
14. Модификаторы уровня доступа (default, public, protected, private).
15. Абстрактные классы и обобщение типов
16. Модификаторы static и final. Модификаторы полей - общие рекомендации
17. Шаблон проектирования Singleton
18. Проектирование абстрактных классов. Вложенные классы
19. Перечислимые типы
20. Java интерфейсы
21. Типы наследования
22. Композиция объектов и делегирование метода
23. Реализация нескольких интерфейсов
24. Шаблон проектирования DAO
25. Настраиваемые классы и параметры типов. Вывод типа (diamond)
26. Коллекции и настраиваемые типы List, set и Map. Stack и Deque
27. Манипулирование строками с помощью StringBuilder и StringBuffer
28. Основные методы String. Разбор текста в Java
29. Обработка ввода с помощью Scanner
30. Вывод текста и форматирование
31. Регулярные выражения с использованием классов Pattern и Matcher
32. Категории исключений. Классы стандартных исключений Java. Создание пользовательских классов исключений
33. Использование предложений try-catch и finally. Использование try-with-resources и интерфейса AutoCloseable. Возможность multi-catch. Обработка исключений Assertions
34. I/O средствами Java. Чтение входного потока консоли. Запись на console. Использование I/O потоков. Цепочки I/O потоков. Канальный I/O
35. Запись и чтение объектов с использованием сериализации
36. Интерфейс Path
37. Класс Files
38. Операции Directory и File
39. Управление атрибутами файловой системы
40. Чтение, запись и создание файлов. Отслеживание

	<p>изменений файловой системы for file system changes</p> <p>41. Планирование выполнения заданий операционной системы</p> <p>42. Многопоточное окружение. Создание многопоточных решений. Общий доступ потоков к данным</p> <p>43. Синхронизация и взаимная блокировка. Неизменяемые объекты</p> <p>44. Создание атомарных переменных. Использование Read-Write блокировок</p> <p>45. Поточно-безопасные коллекции. Конкурентные синхронизаторы (семафоры, фазировщики и др.)</p> <p>46. Исполнители и пулы потоков для конкурентного планирования заданий</p> <p>47. Параллелизм и фреймворк Fork-Join</p> <p>48. Макет JDBC API. JDBC драйверы. Запросы и результаты. PreparedStatement и CallableStatement</p> <p>49. Транзакции</p> <p>50. RowSet 1.1 RowSetProvider и RowSetFactory</p> <p>51. Шаблон DAO и JDBC</p> <p>52. Преимущества локализации. Определение локали. Чтение и установка локали с помощью объекта Locale.</p>
<p><i><b>Имеет практический опыт:</b></i></p> <p>Решения основных задач объектно-ориентированного программирования (ПК-1)</p> <p>Разработки программного обеспечения с использованием различных объектно-ориентированных языков программирования (ПК-3)</p>	<p>Выполнение лабораторных работ:</p> <p style="text-align: center;"><b>3 семестр</b></p> <p>Лабораторная работа 1. Создание простейшего консольного приложения.</p> <p>Лабораторная работа 2. Создание приложения с использованием управляющих конструкций.</p> <p>Лабораторная работа 3. Создания приложения для работы с массивами.</p> <p>Лабораторная работа 4. Создание приложения для работы с циклами.</p> <p>Лабораторная работа 5. Создание конструкторов. Реализация инкапсуляции.</p> <p>Лабораторная работа 6. Создание простого Java-класса.</p> <p style="text-align: center;"><b>4 семестр</b></p> <p>Лабораторная работа 1. Реализация интерфейса. Применение паттерна DAO.</p> <p>Лабораторная работа 2. Подсчет номера деталей с помощью HashMap.</p> <p>Лабораторная работа 3. Разбор текста с помощью split().</p> <p>Лабораторная работа 4. Обработка исключений.</p> <p>Лабораторная работа 5. Написание простой консоли I/O приложения.</p> <p>Лабораторная работа 6. Написание слияние файлов приложения. Рекурсивное копирование. Использование PathMatcher для рекурсивного удаления.</p> <p>Лабораторная работа 7. Синхронизация доступа к разделяемым данным. Реализация многопоточной программы.</p> <p>Лабораторная работа 8. Построение базы данных</p>

## 7.2. Методические рекомендации к определению процедуры оценивания знаний, умений, навыков и (или) опыта деятельности

Рабочая учебная программа дисциплины содержит следующие структурные элементы:

- перечень компетенций, формируемых в результате изучения дисциплины в процессе освоения образовательной программы;
- типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности в процессе освоения образовательной программы (далее—задания). Задания по каждой компетенции, как правило, не должны повторяться.

Требования по формированию задания на оценку ЗНАНИЙ:

- обучающийся должен воспроизводить и объяснять учебный материал с требуемой степенью научной точности и полноты;
- применяются средства оценивания компетенций: тестирование, вопросы по основным понятиям дисциплины и т.п.

Требования по формированию задания на оценку УМЕНИЙ:

- обучающийся должен решать типовые задачи (выполнять задания) на основе воспроизведения стандартных алгоритмов решения;
- применяются следующие средства оценивания компетенций: простые ситуационные задачи (задания) с коротким ответом или простым действием, упражнения, задания на соответствие или на установление правильной последовательности, эссе и другое.

Требования по формированию задания на оценку навыков и (или) опыта деятельности:

- обучающийся должен решать усложнённые задачи (выполнять задания) на основе приобретенных знаний, умений и навыков, с их применением в определенных ситуациях;
- применяются средства оценивания компетенций: задания, требующие многошаговых решений как в известной, так и в нестандартной ситуациях, задания, требующие поэтапного решения и развернутого ответа, ситуационные задачи, проектная деятельность, задания расчетно-графического типа. Средства оценивания компетенций выбираются в соответствии с заявленными результатами обучения по дисциплине.

Процедура выставления оценки доводится до сведения обучающихся в течение месяца с начала изучения дисциплины путем ознакомления их с технологической картой дисциплины, которая является неотъемлемой частью рабочей учебной программы по дисциплине.

В результате оценивания компетенций по дисциплине студенту начисляются баллы по шкале, указанной в рабочей учебной программе по дисциплине.

## 7.3. Описание показателей и критериев оценивания компетенций, описание шкал оценивания

Успешность усвоения дисциплины характеризуется качественной оценкой на основе листа оценки сформированности компетенций, который является приложением к зачетно-экзаменационной ведомости при проведении промежуточной аттестации по дисциплине.

### Критерии оценивания компетенций

Компетенция считается сформированной, если теоретическое содержание курса освоено полностью; при устных собеседованиях студент исчерпывающе, последовательно, четко и логически стройно излагает учебный материал; свободно справляется с задачами, вопросами и другими видами заданий, требующих применения знаний, использует в ответе дополнительный материал; все предусмотренные рабочей учебной программой задания

выполнены в соответствии с установленными требованиями, студент способен анализировать полученные результаты; проявляет самостоятельность при выполнении заданий, качество их выполнения оценено числом баллов от 86 до 100, что соответствует повышенному уровню сформированности компетенции.

Компетенция считается сформированной, если теоретическое содержание курса освоено полностью; при устных собеседованиях студент последовательно, четко и логически стройно излагает учебный материал; справляется с задачами, вопросами и другими видами заданий, требующих применения знаний; все предусмотренные рабочей учебной программой задания выполнены в соответствии с установленными требованиями, студент способен анализировать полученные результаты; проявляет самостоятельность при выполнении заданий, качество их выполнения оценено числом баллов от 61 до 85,9, что соответствует пороговому уровню сформированности компетенции.

Компетенция считается несформированной, если студент при выполнении заданий не демонстрирует знаний учебного материала, допускает ошибки, неуверенно, с большими затруднениями выполняет практические работы, не демонстрирует необходимых умений, доля невыполненных заданий, предусмотренных рабочей учебной программой составляет 55 %, качество выполненных заданий не соответствует установленным требованиям, качество их выполнения оценено числом баллов ниже 61, что соответствует до порогового уровня.

### **Шкала оценки уровня освоения дисциплины**

Качественная оценка может быть выражена: в процентном отношении качества усвоения дисциплины, которая соответствует баллам, и переводится в уровневую шкалу и оценки «отлично» / 5, «хорошо» / 4, «удовлетворительно» / 3, «неудовлетворительно» / 2, «зачтено», «не зачтено». Преподаватель ведет письменный учет текущей успеваемости студента в соответствии с технологической картой по дисциплине.

Шкала оценки результатов освоения дисциплины, сформированности компетенций

<b>Шкалы оценки уровня сформированности компетенции (й)</b>		<b>Шкала оценки уровня освоения дисциплины</b>		
<i>Уровневая шкала оценки компетенций</i>	<i>100 балльная шкала, %</i>	<i>100 балльная шкала, %</i>	<i>5-балльная шкала, дифференцированная оценка/балл</i>	<i>недифференцированная оценка</i>
допороговый	ниже 61	ниже 61	«неудовлетворительно» / 2	Не зачтено
пороговый	61-85,9	61-69,9	«удовлетворительно» / 3	зачтено
		70-85,9	«хорошо» / 4	зачтено
повышенный	86-100	86-100	«отлично» / 5	зачтено

## **8. Учебно-методическое и информационное обеспечение дисциплины**

### **8.1. Перечень основной и дополнительной учебной литературы, необходимой для освоения дисциплины**

#### **Списки основной литературы**

1. Блох, Д. Java. Эффективное программирование [Текст] / Д. Блох ; пер. Е. Коротылев. - 2-е изд. - М. : Лори. - 2015. - 294 с.

2. Васильев, А. Н. Java. Объектно-ориентированное программирование [Текст] : базовый курс по объект.-ориентир. прогр. для магистров и бакалавров / А. Н. Васильев. - СПб. : Питер. - 2014. - 396 с.
3. Фейерштейн, С. Oracle PL/SQL для профессионалов [Текст] / С. Фейерштейн, Б. Прибыл ; [пер. с англ. Е. Матвеев]. - 6-е изд. - СПб. : Питер. - 2015. - 1024 с.
4. Яшин, В. Н. Информатика. Программные средства персонального компьютера [Электронный ресурс] : учеб. пособие для вузов по направлению "Приклад. информатика" и др. экон. специальностям / В. Н. Яшин. - Документ Bookread2. - М. : ИНФРА-М, 2018. - 236 с. - Режим доступа: <http://znanium.com/bookread2.php?book=937489>.

### **Списки дополнительной литературы**

5. Колдаев, В. Д. Основы алгоритмизации и программирования [Электронный ресурс] : учеб. пособие для сред. проф. образования по группе специальностей "Информатика и вычисл. техника" / В. Д. Колдаев ; под ред. Л. Г. Гагариной. - Документ Bookread2. - М. : ФОРУМ [и др.]. - 2015. - 413 с. : ил., табл. - Режим доступа: <http://znanium.com/bookread2.php?book=484837#>.
6. Монахов, В. В. Язык программирования Java и среда NetBeans [Текст] / В. В. Монахов. - 3-е изд. [перераб. и доп.]. - СПб. : БХВ-Петербург. - 2012. - 703 с. : ил. - Библиогр.: с. 698. - В прил. одноимен.
7. Плотникова, Н. Г. Информатика и информационно-коммуникационные технологии (ИКТ) [Электронный ресурс] : учеб. пособие для СПО / Н. Г. Плотникова. - Документ Bookread2. - М. : РИОР [и др.]. - 2017. - 124 с. - Режим доступа: <http://znanium.com/bookread2.php?book=760298>.
8. Программирование [Электронный ресурс] : учеб. для студентов вузов по направлению "Пед. образование" : в 2 т. Т. 1 / Э. А. Нигматулина [и др.] ; под ред. Н. И. Пака. - Документ AdobeAcrobat. - М. : Академия, 2013. - 63,9 МБ, 267 с. - Режим доступа: <http://elib.tolgas.ru>.
9. Программирование [Электронный ресурс] : учеб. для студентов вузов по направлению "Пед. образование" : в 2 т. Т. 2 / Э. А. Нигматулина [и др.] ; под ред. Н. И. Пака. - Документ AdobeAcrobat. - М. : Академия, 2013. - 60,4 МБ, 240 с. : ил. - Режим доступа: <http://elib.tolgas.ru>.
10. Шилдт, Г. Java. Руководство для начинающих [Текст] / Г. Шилдт ; [пер. с англ. и ред. И. В. Берштейна ]. - 5-е изд. - М. : Вильямс. - 2014. - 619 с. - (Oracle)

### **8.2. Перечень ресурсов информационно-телекоммуникационной сети "Интернет" (далее - сеть "Интернет"), необходимых для освоения дисциплины**

#### **Интернет-ресурсы**

1. Инженерное образование [Электронный ресурс] : журнал. - Режим доступа: <http://aeer.ru/ru/magazin.htm>. - Загл. с экрана.
2. Информатизация образования и науки [Электронный ресурс] : журнал. - Режим доступа: <http://www.informika.ru/pachatnye-izdaniya/zhurnal-informatizaciya-obrazovaniya-i-nauki/>. - Загл. с экрана.
3. Наука и образование [Электронный ресурс] : журнал. - Режим доступа: <http://technomag.edu.ru/mdocs/score.html>. - Загл. с экрана.
4. Электронная библиотека. Техническая литература [Электронный ресурс]. - Режим доступа: <http://techliter.ru/>. - Загл. с экрана.
5. Электронная библиотечная система Поволжского государственного университета сервиса [Электронный ресурс]. - Режим доступа: <http://elib.tolgas.ru/>. - Загл. с экрана.
6. Электронно-библиотечная система Znanium.com [Электронный ресурс]. - Режим доступа: <http://znanium.com/>. - Загл. с экрана.

**9. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине, включая перечень программного обеспечения и информационных справочных систем**

Краткая характеристика применяемого программного обеспечения

№ п/п	Программный продукт	Характеристика	Назначение при освоении дисциплины
1	Пакет Microsoft Office	Офисный пакет приложений, созданных корпорацией Microsoft для операционных систем Microsoft Windows	Выполнение и оформление отчетов по лабораторным работам
2	Браузер Internet Explorer	Программа-браузер, разработанная корпорацией Microsoft. Входит в комплект операционных систем семейства Windows.	Поиск и просмотр основной и дополнительной литературы
3	Среда NetBeans	Свободная интегрированная среда разработки приложений (IDE) на языках программирования Java, Python, PHP, JavaScript, C, C++, Ада и ряда других.	Выполнение лабораторных работ

**10. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине**

Для проведения занятий лекционного типа используются специальные помещения - учебные аудитории, укомплектованные специализированной мебелью и техническими средствами обучения, служащими для представления учебной информации.

Для проведения лабораторных работ используются учебные аудитории, оснащенные персональными компьютерами с операционной системой Microsoft Windows; пакетом Microsoft Office; ПО NetBeans; браузером Internet Explorer.

Для текущего контроля и промежуточной аттестации используются специальные помещения - учебные аудитории, укомплектованные специализированной мебелью, и (или) компьютерные классы, оснащенные компьютерной техникой с возможностью подключения к сети Интернет и обеспечением доступа в электронную информационно-образовательную среду университета.

Для самостоятельной работы обучающихся используются специальные помещения - учебные аудитории для самостоятельной работы, оснащенные компьютерной техникой с возможностью подключения к сети Интернет и обеспечением доступа в электронную информационно-образовательную среду университета.



Факультет информационно-технического сервиса  
 кафедра «Информационный и электронный сервис»  
 направление подготовки 09.03.04 «Программная инженерия»  
 направленности (профиля) «Разработка программно-информационных систем»  
**4 семестр**

№	Виды контрольных точек	Кол-во контр. точек	Кол-во баллов за 1 контр. точку	График прохождения контрольных точек																зач. неделя
				Февраль			Март				Апрель				Май					
				4			17	31				14	28	29		12	13			
1	Обязательные:																			
1 . 1	Работа на лабораторных занятиях	7	9				+	+					+	+	+		+	+		
2	Творческий рейтинг:																			
2 . 1	Участие в конференциях	1	7										+							
2 . 2	Индивидуальное задание	1	10										+							
2 . 3	Участие в научной работе	1	20										+							
	Итого																			
	Форма контроля												к.т						Экзамен	



