

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Выборнова Любовь Алексеевна

Должность: Ректор

Дата подписания: 03.02.2022 15:17:47

Уникальный программный идентификатор:

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 «Прикладное программное обеспечение»

для студентов направления подготовки
09.03.02 «Информационные системы и технологии»
направленности (профиля) Информационные системы и технологии»

Тольятти 2018 г.

Рабочая учебная программа по дисциплине «Прикладное программное обеспечение» включена в основную профессиональную образовательную программу направления подготовки 09.03.02 «Информационные системы и технологии» направленности (профиля) «Информационные системы и технологии» решением Президиума Ученого совета

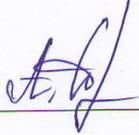
Протокол № 4 от 28.06.2018 г.

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

Рабочая учебная программа по дисциплине «Прикладное программное обеспечение» разработана в соответствии с Федеральным государственным образовательным стандартом направления подготовки 09.03.02 «Информационные системы и технологии» утвержденный приказом Министерства образования и науки Российской Федерации от 12 марта 2015 г. N 219.

Составил: к.т.н., доцент А.А. Попов

СОГЛАСОВАНО:

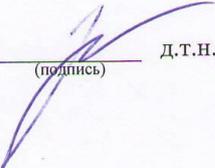
Директор научной библиотеки  В.Н.Еремина

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 «Прикладное программное обеспечение», соотнесенных с планируемыми результатами освоения образовательной программы

1.1. Цели освоения дисциплины «Прикладное программное обеспечение» является приобретение теоретических знаний и закрепление профессиональных навыков по разработке прикладного программного обеспечения при решении задач в различных сферах деятельности.

1.2. В соответствии с видами профессиональной деятельности, на которые ориентирована образовательная программа указанного направления подготовки, содержание дисциплины «Прикладное программное обеспечение»:

сервисно-эксплуатационная деятельность.

- адаптация приложений к изменяющимся условиям функционирования.

1.3. Компетенции обучающегося, формируемые в результате освоения дисциплины

В результате освоения дисциплины у обучающихся формируются следующие компетенции:

Код компетенции	Наименование компетенции
ПК-32	Способность адаптировать приложения к изменяющимся условиям функционирования

1.4. Перечень планируемых результатов обучения по дисциплине

Результаты освоения дисциплины	Технологии формирования компетенции по указанным результатам	Средства и технологии оценки по указанным результатам
<p><i>Знает: ПК-32</i> -технологию разработки алгоритмов и программ, методы отладки и решения задач на ЭВМ в различных режимах; -состав и структуру инструментальных средств и систем программирования, тенденции их развития (операционные системы, языки программирования, технические средства).</p>	Лекции	Собеседование
<p><i>Умеет: ПК-32</i> -ставить задачу и разрабатывать алгоритм ее решения, использовать прикладные системы программирования, разрабатывать основные программные документы; - разрабатывать программные приложения и формулировать требования к используемым компонентам информационных систем процесса предоставления услуг, в том числе в соответствии с требованиями потребителя, на основе новейших информационных и коммуникационных технологий.</p>	Лабораторные работы	Собеседование Защита лабораторных работ
<p><i>Имеет практический опыт: ПК-32</i> -работы с инструментальными средствами и системами программирования предметной области, прикладных и вычислительных процессов; -работы с языками процедурного и объектно-ориентированного программирования.</p>	Лекции Лабораторные работы	Защита лабораторных работ

2. Место дисциплины в структуре образовательной программы

Дисциплина относится к вариативной части.

Ее освоение осуществляется в 6 (очная форма) / 7 (очно-заочная форма, заочная форма) семестре.

№ п/п	Наименование дисциплин, определяющих междисциплинарные связи	Код компетенции(й)
	Предшествующие дисциплины	
1	Программирование	ПК-12
2	Операционные системы	ПК-30, ПК-32
3	Информационные технологии	ОПК-1, ОПК-4
	Последующие дисциплины и практики	
4	Методы и средства проектирования информационных систем и технологий	ПК-11, ПК-33
5	Технологии сети Internet	ПК-11
6	Защита информации	ПК-31
7	Корпоративные информационные системы	ПК-17
8	Преддипломная практика	ПК-11, ПК-13, ПК-14, ПК-16, ПК-17, ПК-31, ПК-32

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	очно-заочная форма обучения	заочная форма обучения
Итого часов	216 ч.	216 ч.	216 ч.
Зачетных единиц	6 з.е.	6 з.е.	6 з.е.
Лекции (час)	28	8	8
Практические (семинарские) занятия (час)	-	-	-
Лабораторные работы (час)	48	14	14
Самостоятельная работа (час)	113	185	185
Курсовой проект (работа) (+,-)	КП	КП	КП
Контрольная работа (+,-)	-	-	-
Экзамен, семестр /час.	6/27 ч	7/9 ч	7/9 ч
Зачет (дифференцированный зачет), семестр	-	-	-
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	
1	<p>Тема 1. Языки программирования и интегрированные среды разработки программного обеспечения. Прикладное и системное программное обеспечение. Объектно-ориентированное программирование. Среда Visual Studio.Net. Язык C#, основные понятия языка: состав языка, типы данных.</p>	2/0,5/0,5	-/-/-	4/1/1	6/12/12	Конспект, защита лабораторных работ
2	<p>Тема 2 Переменные, операции и выражения. Переменные. Именованные константы. Операции и выражения. Линейные программы.</p>	2/0,5/0,5	-/-/-		9/13/13	Конспект, защита лабораторных работ
3	<p>Тема 3 Операторы. Выражения, блоки и пустые операторы. Операторы ветвления. Операторы цикла. Операторы передачи управления. Обработка исключительных ситуаций, операторы try, throw, checked и unchecked. Класс Exception</p>	2/0,5/0,5	-/-/-	8/3	9/17/17	Конспект, защита лабораторных работ
4	<p>Тема 4 Классы: основные понятия. Присваивание и сравнение объектов. Данные: поля и константы. Методы. Параметр this. Конструкторы. Свойства.</p>	2/0,5/0,5	-/-/-	4/1	8/13/13	Конспект, защита лабораторных работ
5	<p>Тема 5 Массивы и строки. Массивы. Оператор foreach. Массивы объектов. Символы и строки. Класс Random.</p>	2/0,5/0,5	-/-/-	12/2	8/21/21	Конспект, защита лабораторных работ
6	<p>Тема 6 Классы: подробности. Перегрузка методов. Рекурсивные методы. Методы с переменным количеством аргументов. Метод Main. Индексаторы. Операции класса. Деструкторы. Вложенные типы.</p>	2/0,5/0,5	-/-/-	4/1/1	9/14/14	Конспект, сообщение, опрос на лекции
7	<p>Тема 7 Иерархии классов. Наследование. Виртуальные методы. Абстрактные классы. Бесплодные классы. Класс object.</p>	2/0,5/0,5	-/-/-	4/1/1	8/14/14	Конспект, сообщение, опрос на лекции
8	<p>Тема 8 Интерфейсы и структурные типы. Синтаксис интерфейса. Реализация интерфейса. Работа с объектами через</p>	2/0,5/0,5	-/-/-	4/1/1	9/13/13	Конспект, защита лабораторных работ

№ 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	
	интерфейсы. Операции is и as.					
9	Тема 9 Делегаты, события и потоки выполнения. Делегаты. События. Многопоточные приложения. Рекомендации по программированию.	2/0,5/0,5	-/-/-	-/-/-	8/10/10	Конспект
10	Тема 10 Работа с файлами. Потоки байтов. Асинхронный ввод-вывод. Потоки символов. Двоичные потоки. Консольный ввод-вывод. Работа с каталогами и файлами. Сохранение объектов (сериализация).	2/0,5/0,5	-/-/-	-/-/-	8/10/10	Конспект
11	Тема 11 Сборки, библиотеки, атрибуты, директивы. Сборки. Создание библиотеки. Использование библиотеки. Рефлексия. Атрибуты. Пространства имен. Директивы препроцессора.	2/0,5/0,5	-/-/-	-/-/-	8/12/12	Конспект
12	Тема 12 Структуры данных, коллекции и классы-прототипы. Абстрактные структуры данных. Пространство имен System.Collections. Классы-прототипы. Частичные типы. Обнуляемые типы.	2/0,5/0,5	-/-/-	4/2/2	8/13/13	Конспект, защита лабораторных работ
13	Тема 13 Программирование под Windows. Событийно-управляемое программирование. Шаблон Windows-приложения. Класс Control. Элементы управления. Предварительные замечания о формах. Класс Form. Диалоговые окна. Класс Application. Краткое введение в графику.	2/1	-/-/-	4/2/2	9/13/13	Конспект, защита лабораторных работ
14	Тема 14 Дополнительные средства C#. Заключение. Небезопасный код. Регулярные выражения. Документирование в формате XML. Заключение.	2/1/1	-/-/-	-/-/-	6/10/10	Конспект
	Промежуточная аттестация по дисциплине	28/8/8	-/-/-	48/14/14	113/185/185	Экзамен

Примечание:

-/-, объем часов соответственно для очной, заочной форм обучения

4.2. Содержание практических (семинарских) занятий

Практические работы планом не предусмотрены.

4.3. Содержание лабораторных работ

№	Наименование темы лабораторных занятий	Объем часов	Наименование темы дисциплины
1	Занятие 1. «Линейные программы»	4/1/1	Тема 1. Языки программирования и интегрированные среды разработки программного обеспечения.
2	Занятие 2. «Разветвляющиеся вычислительные процессы. Организация циклов»	8/3/3	Тема 3. Операторы.
3	Занятие 3. «Простейшие классы»	4/1/1	Тема 4. Классы: основные понятия.
4	Занятие 4. «Одномерные массивы. Двумерные массивы. Строки»	12/2/2	Тема 5. Массивы и строки.
5	Занятие 5. «Классы и операции»	4/1/1	Тема 6. Классы: подробности.
6	Занятие 6. «Наследование»	4/1/1	Тема 7. Иерархии классов.
7	Занятие 7. «Структуры»	4/1/1	Тема 8. Интерфейсы и структурные типы.
8	Занятие 8. «Интерфейсы и параметризованные коллекции»	4/2/2	Тема 12. Структуры данных, коллекции и классы-прототипы.
9	Занятие 9. «Создание Windows-приложений»	4/2/2	Тема 13. Программирование под Windows.
	Итого	48/14/14	

Примечание:

-/-/-, объем часов соответственно для очной, очно-заочной, заочной форм 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
ПК-32	Выполнение индивидуальных заданий в виде доклада и	Доклад, презентация	Собеседование	113/185/185

	презентации на заданную тему.		
			Итого
			113/185/ 185

Примечание:

-/-/, объем часов соответственно для очной, очно-заочной, заочной форм обучения

Рекомендуемая литература:

1. Гуриков, С. Р. Основы алгоритмизации и программирования на Python [Электронный ресурс] : учеб. пособие для вузов по направлениям подгот. 09.03.01 "Информатика и вычисл. техника", 09.03.02 "Информ. системы и технологии" (квалификация (степень) "бакалавр") / С. Р. Гуриков. - Документ Bookread2. - М. : ФОРУМ [и др.], 2017. - 342 с. - Режим доступа: <http://znanium.com/bookread2.php?book=772265>

2. Залогова, Л. А. Основы объектно-ориентированного программирования на базе языка С# [Электронный ресурс] : учеб. пособие / Л. А. Залогова. - Документ Reader. - СПб. [и др.] : Лань, 2018. - 191 с. - Режим доступа: <https://e.lanbook.com/reader/book/106731/#1>.

3. Кузин, А. В. Программирование на языке Си [Электронный ресурс] : [учеб. пособие для вузов] / А. В. Кузин, Е. В. Чумакова. - Документ Bookread2. - М. : ФОРУМ, 2015. - 142 с. - Библиогр.: с. 139. - ([Высшее образование]). - Режим доступа: <http://znanium.com/bookread2.php?book=505194>

4. Хорев, П. Б. Объектно-ориентированное программирование с примерами на С# [Электронный ресурс] : учеб. пособие для вузов по направлениям 01.03.02 "Приклад. математика и информатика" и 09.03.01 "Информатика и вычисл. техника" / П. Б. Хорев. - Документ Bookread2. - М. : ФОРУМ [и др.], 2016. - 200 с. - Режим доступа: <http://znanium.com/bookread2.php?book=529350>.

Содержание заданий для самостоятельной работы

Темы рефератов (письменных работ, эссе, докладов и т.п.)

1. Разработка приложений с параллельным выполнением операций на базе платформы .NET.
2. Особенности разработки приложений для iOS.
3. Особенности разработки приложений на платформе 1С:Предприятие 8.
4. Особенности разработки приложений на XNA Framework.
5. Особенности разработки приложений для мобильных устройств на платформе Windows Phone.
6. Особенности разработки Windows Store приложений.
7. Особенности разработки приложений для Windows Phone 8.
8. Сравнение приложений на С# и Ruby.
9. Сравнение приложений на С# и Java.
10. Сравнение скорости вычислений на JVM и MSIL.

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
Компьютерные симуляции	-	-	№ 1-5
Разбор конкретных ситуаций	-	-	№ 6-9
Слайд-лекции	№ 1-14	-	-

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

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).

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

Лабораторные работы

№	Наименование лабораторных работ	Задание по лабораторным работам
1	Линейные программы	Постановка задачи; краткие теоретические сведения; работа с переменными, именованными константами, с операциями и выражениями, линейными программами на языке C#.
2	Разветвляющиеся вычислительные процессы. Организация циклов	Постановка задачи; краткие теоретические сведения. Написание программ с выражениями, блоками и пустыми операторами. Написание программ с операторами ветвления, операторами цикла, операторами передачи управления. Написание программ с обработкой исключительных ситуаций. Работа с классами Exception.
3	Простейшие классы	Постановка задачи; краткие теоретические сведения. Написание программ с присваиванием и сравнением объектов. Работа с данными: поля и константы. Работа с методами. Работа с параметрами this. Работа с конструкторами. Работа со свойствами.
4	Одномерные массивы. Двумерные массивы. Строки	Постановка задачи; краткие теоретические сведения; Написание программ с массивами. Работа с оператором foreach. Написание программ с массивами объектов.

№	Наименование лабораторных работ	Задание по лабораторным работам
		Написание программ с символами и строками. Работа с классом Random.
5	Классы и операции	Постановка задачи; краткие теоретические сведения. Написание программ с перегрузкой методов. Написание программ с рекурсивными методами, методами с переменным количеством аргументов, методом Main. Работа с индексаторами. Работа с операциями классов, деструкторами, вложенными типами.
6	Наследование	Постановка задачи; краткие теоретические сведения. Работа с наследованием. Работа с виртуальными методами. Работа с абстрактными классами, бесплодными классами, классом object.
7	Структуры	Постановка задачи; краткие теоретические сведения. Реализация интерфейса. Работа с объектами через интерфейсы. Работа с операциями is и as. Работа с интерфейсами и наследованием. Работа со стандартными интерфейсами .NET. Написание программ со структурами и перечислениями.
8	Интерфейсы и параметризованные коллекции	Постановка задачи; краткие теоретические сведения. Работа с абстрактными структурами данных. Реализация классов-прототипов. Написание программ с частичными типами и обнуляемыми типами.
9	Создание Windows-приложений	Постановка задачи; краткие теоретические сведения. Создание Windows-приложений.

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 проектов

В ходе изучения дисциплины студентами выполняется курсовой проект.

Целью курсового проекта по дисциплине «Прикладное программное обеспечение» является систематизация, закрепление и расширение полученных теоретических знаний, практических навыков и умений для творческого применения в решении конкретных задач в области прикладного программного обеспечения. Курсовой проект призван также

расширить объем знаний студентов в данной области и создать реальную основу использования этих знаний для решения задач при изучении последующих дисциплин и в будущей практической деятельности.

Задачей курсового проектирования является самостоятельное изучение, анализ и решение вопросов, связанных с разработкой прикладного программного продукта.

В процессе курсового проектирования студент работает с источниками в Internet, с электронной библиотекой и специальной литературой по предметной области, использует справочники и общесистемные методические материалы. Студент должен проявить свои навыки к самостоятельной работе с научно-технической литературой, к обобщению накопленного опыта и свое умение делать научно обоснованные выводы и рекомендации.

В качестве задания на курсовое проектирование студенту предлагается создать прикладной программный продукт (приложение), решающий некоторую поставленную задачу. Результатом решения является:

- 1) созданное приложение;
- 2) пояснительная записка, составленная с учетом требования стандартов университета.

Для выполнения курсового проектирования студенту назначается руководитель курсового проектирования. Как правило, руководителем является преподаватель той дисциплины, в рамках которой выполняется курсовой проект. Студент выбирает и согласует с руководителем проектирования тему проекта. Этот процесс заканчивается тем, что студент и руководитель подписывают Задание на проектирование, утвержденное заведующим кафедрой. Задание включает, помимо названия темы, требования к разрабатываемому приложению, конкретизированное содержание пояснительной записки и перечень графического материала.

Студенты заочной формы обучения получают задания на кафедре или в деканате. Пакет заданий для студентов-заочников руководитель готовит ежегодно к началу семестра, в котором выполняется проектирование. Студент заочного отделения выбирает тему проекта по номеру зачетной книжки либо согласует тему индивидуально. Приветствуется выполнение студентом-заочником проекта по тематике предприятия, на котором он работает.

Примеры тем курсовых проектов:

1. Разработать приложение, которое считывает текст из задаваемого пользователем файла и выводит на экран и в файл `Rezult.txt` предложения, содержащие введенное с клавиатуры слово.

2. Разработать приложение, которое считывает текст из задаваемого пользователем файла и выводит на экран и в файл `Rezult.txt` предложения, содержащие даты в формате `dd.mm.yyyy`.

3. Разработать приложение, которое считывает текст из задаваемого пользователем файла и выводит на экран и в файл `Rezult.txt` переданный ему текст, удалив лидирующие пробелы в каждой строке исходного текста.

4. Разработать приложение, которое считывает текст из файла и выводит на экран и в файл `Rezult.txt` фрагменты текста на английском языке.

5. Разработать приложение, которое считывает текст из задаваемого файла и выводит на экран и в файл `Rezult.txt` содержащиеся в тексте цитаты - предложения, заключенные в кавычки.

6. Разработать приложение, которое считывает текст из файла и выводит на экран и в файл `Rezult.txt` предложения, содержащие максимальное количество знаков пунктуации.

7. Разработать приложение, которое считывает текст из файла и выводит на экран и в файл `Rezult.txt` предложения, содержащие заданный знак пунктуации в заданном количестве (3 запятых, двоеточие или `!?` и т.д.).

8. Разработать приложение, которое считывает текст из файла и определяет, сколько в нем слов, состоящих из заданного пользователем количества символов.

9. Разработать приложение, которое считывает текст из файла, находит самое

длинное слово и определяет, сколько раз оно встретилось в тексте.

10. Разработать приложение, которое считывает текст из файла и выводит на экран и в файл Rezult.txt предложения, состоящие из заданного количества слов.

7. Фонд оценочных средств для проведения текущего контроля успеваемости и промежуточной аттестации обучающихся по дисциплине (дифференцированный зачет)

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

Код оцениваемой компетенции (или ее части)	Тип контроля	Вид контроля	Количество элементов
ПК-32	текущий	устный опрос	55
ПК-32	промежуточный	тест	99

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

Результаты освоения дисциплины	Оценочные средства (перечень вопросов, заданий и др.)
<p>Знает: ПК-32 -технологию разработки алгоритмов и программ, методы отладки и решения задач на ЭВМ в различных режимах; -состав и структуру инструментальных средств и систем программирования, тенденции их развития (операционные системы, языки программирования, технические средства).</p>	<ol style="list-style-type: none"> К типам приложений .NET Framework следует отнести <ul style="list-style-type: none"> - консольные приложения - Windows-приложения - макромедийные приложения Какой метод должна содержать каждая программа на языке C# (С шарп) <ul style="list-style-type: none"> - Main () - Void () - Class () Проверка безопасности выполнения кода носит название <ul style="list-style-type: none"> - верификация - компиляция - модуляция Какие типы файлов могут быть описаны в заголовке PE32 <ul style="list-style-type: none"> - GUI - DLL - CSW К базовым понятиям объектно-ориентированного программирования следует отнести <ul style="list-style-type: none"> - класс - инкапсуляцию - модуль Функции, предназначенные для обработки внутренних данных объекта данного класса, носят название <ul style="list-style-type: none"> - методы - маркеры - типы Совокупность правил доступа к объекту носит название <ul style="list-style-type: none"> - интерфейс

Результаты освоения дисциплины	Оценочные средства (перечень вопросов, заданий и др.)
	<ul style="list-style-type: none"> - модуль - контейнер 8. С механизмом виртуальных методов связываются понятия - полиморфизма - инкапсуляции - полифонизма 9. Совокупность допустимых в языке символов носит название - алфавит - контейнер - метастроку 10. Структура, используемая для хранения элементов по принципу: первым пришел - последним ушел, носит название - стек - модуль - контейнер 11. Какие из приведенных ниже записей соответствуют типам операций C# - комплексные - дуплексные - конструктивные 12. Какая операция используется в C# для создания нового объекта - new - object - create 13. Операция деления с остатком в C# имеет вид - % - & - ! 14. Для организации многократно повторяющихся вычислений используются операторы - цикла - ветвления - наследования 15. К операторам безусловного перехода C# следует отнести - goto - break - depend 16. Функциональный элемент класса, который реализует вычисления, носит название - метод - свойство - маркер 17. К необязательным элементам синтаксиса описания метода следует отнести - атрибуты - спецификаторы - имя метода 18. Переменные или константы, которые получают значения, передаваемые методу при вызове, носят название

Результаты освоения дисциплины	Оценочные средства (перечень вопросов, заданий и др.)
	<ul style="list-style-type: none"> - параметры - методы - контейнеры 19. Для обмена информацией между вызывающим и вызываемым методами используются <ul style="list-style-type: none"> - параметры - процедуры - вызовы 20. Если метод вызывает сам себя в качестве вспомогательного, он называется <ul style="list-style-type: none"> - рекурсивным - циклическим - комплексным 21. В какой рекурсии метод вызывает себя в качестве вспомогательного через другой вспомогательный метод <ul style="list-style-type: none"> - в косвенной - в динамической - в статической 22. К стандартным исключениям следует отнести <ul style="list-style-type: none"> - деление на ноль - попадание вне диапазона определения индекса - инверсию операторов присваивания 23. В C# исключения представляются <ul style="list-style-type: none"> - классами - методами - объектами 24. Тип исключения должен совпадать с типом, заданным <ul style="list-style-type: none"> - в catch-инструкции - в функции Main - в JIT-компиляторе 25. В C# массивы реализованы в виде <ul style="list-style-type: none"> - объектов - типов - ссылок 26. К какому типу данных относятся массивы C#? <ul style="list-style-type: none"> - ссылочному - структурному - модульному 27. Массивы C# реализованы на основе базового класса <ul style="list-style-type: none"> - Array - Object - Struct 28. Какое свойство позволяет определять количество элементов в массиве <ul style="list-style-type: none"> - Length - Width - Size 29. Каждая строка ступенчатого массива фактически является <ul style="list-style-type: none"> - одномерным массивом - динамическим массивом - циклическим массивом

Результаты освоения дисциплины	Оценочные средства (перечень вопросов, заданий и др.)
	<p>30. Регулярные выражения в C# носят название</p> <ul style="list-style-type: none"> - Regex - Defex - Strex <p>31. Из приведенных ниже записей выделите методы класса Char</p> <ul style="list-style-type: none"> - IsUpper - IsControl - IsModule <p>32. Какие из приведенных ниже записей соответствуют методам класса String</p> <ul style="list-style-type: none"> - Concat - Join - Replace - ToString <p>33. Чтобы создать строку, которую можно изменять, в C# предусмотрен класс</p> <ul style="list-style-type: none"> - StringBuilder - StringConstruct - StringSelect <p>34. Регулярные выражения предназначены для обработки</p> <ul style="list-style-type: none"> - текстовой информации - графической информации - табличных данных <p>35. В какие источники можно записывать данные с помощью потоков</p> <ul style="list-style-type: none"> - принтер - экран - файл - фрейм <p>36. Для добавления данных в конец файла используется параметр</p> <ul style="list-style-type: none"> - FileMode.Append - FileMode.CreateNew - FileMode.Truncate <p>37. Из приведенных ниже записей выделите исключения, которые могут генерироваться, если попытка открыть файл оказалась неуспешной</p> <ul style="list-style-type: none"> - ArgumentException - ArgumentNullException - ArgumentOutOfRangeException <p>38. Если достигнут конец файла, то метод ReadByte() возвращает значение</p> <ul style="list-style-type: none"> - -1 - 0 - 1 <p>40. В C# символы реализуются кодировкой</p> <ul style="list-style-type: none"> - Unicode - CHR-80 - ASCII <p>41. Работать с каталогами файловой системы компьютера</p>

Результаты освоения дисциплины	Оценочные средства (перечень вопросов, заданий и др.)
	<p>можно и при помощи класса</p> <ul style="list-style-type: none"> - Directory - File - Parent <p>42. Доступ к физическим файлам можно получать через статические методы класса</p> <ul style="list-style-type: none"> - File - Name - Space <p>43. Класс содержит данные, которые определяют</p> <ul style="list-style-type: none"> - свойства объекта - методы объекта - модификаторы объекта <p>44. Какие функциональные элементы может содержать класс</p> <ul style="list-style-type: none"> - конструкторы - индексаторы - деструкторы - идентификаторы <p>45. Конструктор экземпляра вызывается</p> <ul style="list-style-type: none"> - автоматически - по селектору - по идентификатору <p>46. Какой метод вызывается сборщиком мусора непосредственно перед удалением объекта из памяти</p> <ul style="list-style-type: none"> - деструктор - модификатор - анализатор <p>47. Класс, который наследуется, называется</p> <ul style="list-style-type: none"> - базовым - исходным - контекстным <p>48. В конструкторе базового класса для инициализации полей используется параметр</p> <ul style="list-style-type: none"> - this - mode - reset <p>49. Абстрактный метод является</p> <ul style="list-style-type: none"> - виртуальным - контекстным - модификативным <p>50. Коллекции разработаны на основе</p> <ul style="list-style-type: none"> - интерфейсов - методов - объектов <p>51. Последовательность элементов одного типа носит название</p> <ul style="list-style-type: none"> - список - массив - модуль <p>52. Очередь реализует принцип обслуживания</p> <ul style="list-style-type: none"> - FIFO

Результаты освоения дисциплины	Оценочные средства (перечень вопросов, заданий и др.)
	<ul style="list-style-type: none"> - LIFO - FILO <p>53. Какое свойство позволяет узнать или установить емкость вызывающего динамического массива типа ArrayList</p> <ul style="list-style-type: none"> - Capacity - Marker - Length <p>54. Вызов метода или делегата осуществляется в С# с помощью операции</p> <ul style="list-style-type: none"> - x() - x{ } - x[] <p>55. CLR - это</p> <ul style="list-style-type: none"> - Общезыковая объектно-ориентированная среда выполнения - Динамическая платформа обратной связи - Метод комплексной обработки данных
<p>Умеет: ПК-32 -ставить задачу и разрабатывать алгоритм ее решения, использовать прикладные системы программирования, разрабатывать основные программные документы; - разрабатывать программные приложения и формулировать требования к используемым компонентам информационных систем процесса предоставления услуг, в том числе в соответствии с требованиями потребителя, на основе новейших информационных и коммуникационных технологий.</p>	<ol style="list-style-type: none"> 1. Какие компоненты включает в себя платформа .NET Framework? 2. Опишите семь функций общезыковой среды выполнения программ CLR. 3. Что определяет следующий синтаксис: [атрибуты] [модификаторы] enum [: базовый тип] { список-перечисления констант (через запятую) }; 4. Опишите четыре наиболее часто встречающиеся типы приложений. 5. Поясните понятия "класс", "объект", "интерфейс", "инкапсуляция", "наследование", "поли- морфизм", "событие" ООП. 6. Чем отличаются соглашения о правилах создания имен - нотации Pascal, венгерская нотация и нотация Camel? 7. Опишите набор базовых типов языка С#. 8. Опишите разницу между размерными типами (типами по значению) и ссылочными типами данных. 9. Как организован ввод-вывод данных в языке С#? 10. Опишите назначение и применение методов WriteLine и ReadLine, реализованных в классе Console. 11. Приведите примеры операций явного преобразования типа. 12. Поясните как выполняется неявное (без участия программистов) преобразование типов в выражениях. 13. В каких случаях применяются составные операторы (блоки)? Какова область видимости блока? 14. Опишите синтаксис: [атрибуты] [спецификторы] тип_возвращаемого_результата имя_метода ([список_параметров]) { тело_метода; return значение } 15. Что называется перегрузкой методов? 16. Дайте определение и пример рекурсивного метода. 17. Поясните различие между ошибкой в программе, ошибочной ситуацией и исключительной ситуаций.

Результаты освоения дисциплины	Оценочные средства (перечень вопросов, заданий и др.)
	<p>Приведите примеры.</p> <p>18. Для чего служит подсистема обработки исключений?</p> <p>19. На чем основывается управление обработкой исключений в С#?</p> <p>20. Поясните следующий синтаксис и приведите пример его использования:</p> <pre>try { ... } catch { ... } finally { ... }</pre> <p>21. К каким типам данных – ссылочным или размерным – относятся С#-массивы? Назовите типы массивов в С#.</p> <p>22. Дайте определение и приведите пример объявления ступенчатого массива. В каком виде ступенчатый массив хранится в памяти?</p> <p>23. Поясните следующий синтаксис и приведите пример его использования: тип <code>[][]</code> имя_массива;</p> <p>24. Поясните следующий синтаксис и приведите пример его использования:</p> <pre>foreach (<тип><имя> in <группа>) <тело цикла></pre> <p>25. Сравните методы работы с символами <code>char</code>, неизменяемыми строками <code>string</code>, изменяемыми строками <code>StringBuider</code>.</p> <p>26. В какой кодировке хранятся символы типа <code>char</code>? Какие возможности предоставляют основные статические методы, определенные в классе <code>Char</code> библиотеки <code>.Net</code>?</p> <p>27. Поясните действие оператора</p> <pre>char[] a={ 'm', 'a', 'X', 'i', 'M', 'u', 'S', '!', '!', '!' };</pre> <p>28. Сравните действие операторов:</p> <pre>string s=new string (' ', 20); StringBuilder c = new StringBuilder(20);</pre> <p>29. Для чего предназначены регулярные выражения, какие возможности они предоставляют?</p> <p>30. Для чего предназначены методы <code>IsMatch</code>, <code>Match</code>, <code>Matches</code> класса <code>Regex</code>?</p> <p>31. Как преобразовать байтовый поток в символьный?</p> <p>32. Сравните действие операторов: <code>FileStream fileOut = new FileStream("newText.txt", FileMode.Create, FileAccess.Write); StreamWriter fileOut=new StreamWriter("t.txt", true);</code></p> <p>33. Поясните действие оператора:</p> <p>34. <code>BinaryWriter fOut=new BinaryWriter(new FileStream("t.dat",FileMode.Create));</code></p> <p>35. Приемником стандартного выходного потока по умолчанию является консоль. Как перенаправить поток в файл?</p> <p>36. Источник стандартного входного потока по умолчанию – клавиатура. Как сделать источни- ком потока файл?</p> <p>37. В чем отличие классов <code>Directory</code> и <code>DirectoryInfo</code>. Для чего предназначены эти классы?</p>

Результаты освоения дисциплины	Оценочные средства (перечень вопросов, заданий и др.)
	38. Сравните классы FileSystemInfo, FileInfo и File. Для чего предназначены эти классы? 39. Опишите спецификатор класса – static. 40. Опишите функциональные элементы, которые может содержать класс: данные, методы, конструкторы, свойства, деструкторы, индексаторы, операции, события, типы. 41. Сравните конструкторы класса и конструкторы экземпляра класса. 42. Сохранение объектов (сериализация) в C#. 43. Опишите составные части сборки, как результата компиляции в среде .Net. 44. Создание и использование библиотеки .Net.
Имеет практический опыт: ПК-32 -работы с инструментальными средствами и системами программирования предметной области, прикладных и вычислительных процессов; -работы с языками процедурного и объектно-ориентированного программирования.	Линейные программы Разветвляющиеся вычислительные процессы. Организация циклов Простейшие классы Одномерные массивы. Двумерные массивы. Строки Классы и операции Наследование Структуры Интерфейсы и параметризованные коллекции Создание Windows-приложений

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 усложненные 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 *допороговому уровню*.

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

Шкалы оценки уровня	Шкала оценки уровня освоения дисциплины
----------------------------	--

сформированности компетенции (й)				
Уровневая шкала оценки компетенций	100 бальная шкала, %	100 бальная шкала, %	5-бальная шкала, дифференцированная оценка/балл	Недифференцированная оценка
допороговый	ниже 61	ниже 61	«неудовлетворительно» / 2	Не зачтено
пороговый	61-85,9	70-85,9	«хорошо» / 4	зачтено
		61-69,9	«удовлетворительно» / 3	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. Гуриков, С. Р. Основы алгоритмизации и программирования на Python [Электронный ресурс] : учеб. пособие для вузов по направлениям подгот. 09.03.01 "Информатика и вычисл. техника", 09.03.02 "Информ. системы и технологии" (квалификация (степень) "бакалавр") / С. Р. Гуриков. - Документ Bookread2. - М. : ФОРУМ [и др.], 2017. - 342 с. - Режим доступа: <http://znanium.com/bookread2.php?book=772265>

2. Залогова, Л. А. Основы объектно-ориентированного программирования на базе языка С# [Электронный ресурс] : учеб. пособие / Л. А. Залогова. - Документ Reader. - СПб. [и др.] : Лань, 2018. - 191 с. - Режим доступа: <https://e.lanbook.com/reader/book/106731/#1>.

3. Кузин, А. В. Программирование на языке Си [Электронный ресурс] : [учеб. пособие для вузов] / А. В. Кузин, Е. В. Чумакова. - Документ Bookread2. - М. : ФОРУМ, 2015. - 142 с. - Библиогр.: с. 139. - ([Высшее образование]). - Режим доступа: <http://znanium.com/bookread2.php?book=505194>

4. Хорев, П. Б. Объектно-ориентированное программирование с примерами на С# [Электронный ресурс] : учеб. пособие для вузов по направлениям 01.03.02 "Приклад. математика и информатика" и 09.03.01 "Информатика и вычисл. техника" / П. Б. Хорев. - Документ Bookread2. - М. : ФОРУМ [и др.], 2016. - 200 с. - Режим доступа: <http://znanium.com/bookread2.php?book=529350>.

Списки дополнительной литературы

5. Задачи по программированию [Текст] / С. М. Окулов [и др.] ; под ред. С. М. Окулова. - М. : БИНОМ. Лаб. знаний, 2006. - 820 с.

6. Златопольский, Д. М. Сборник задач по программированию [Текст] / Д. М. Златопольский. - 2-е изд. - СПб. : БХВ-Петербург, 2007. - 240 с. : ил.

7. 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. DVD. - Предм. указ..

8. Монахов, В. В. Язык программирования Java и среда NetBeans [Электронный ресурс] : DVD / В. В. Монахов. - 3-е изд. - СПб. : БХВ-Петербург, [б.г.]. - 1,72 Гб. - DVD. - Прил. к одноимен. кн..

9. Монган, Д. Работа мечты для программиста. Тестовые задачи и вопросы при собеседовании в ведущих IT-компаниях [Текст] / Д. Монган, Н. Киндлер, Э. Гижере. - 3-е изд. - СПб. : Питер, 2014. - 368 с. : ил., табл. - (Для профессионалов)

10. Нортроп, Т. Основы разработки приложений на платформе Microsoft .NET Framework [Текст] : [учеб. курс + CD] / Т. Нортроп, Ш. Уилдермьюс, Б. Райан ; пер. с англ.

под общ. ред. А. Е. Соловченко. - М. [и др.] : Рус. ред. [и др.], 2007. - XXI, 842 с. : ил. - В прил.: одноимен. CD-ROM на англ. яз. - Слов. терминов. - (Официальное пособие для самоподготовки. Учебный курс Microsoft)

11. Окулов, С. М. Основы программирования [Текст] / С. М. Окулов. - М. : Лаб. Базовых Знаний, 2002. - 424 с. : ил.

12. Щупак, Ю. А. Win32 API. Разработка приложений для Windows [Текст] / Ю. А. Щупак. - СПб. : Питер, 2008. - 592 с. : ил. - Библиогр.: с. 587. - Алф. указ.. - (Библиотека программиста)

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

Интернет-ресурсы

1. ИНТУИТ. Национальный Открытый Университет [Электронный ресурс]. – Режим доступа: <http://www.intuit.ru/>. – Загл. с экрана.
2. Образовательные ресурсы Интернета. Информатика [Электронный ресурс]. - Режим доступа: <http://www.alleng.ru/edu/comp.htm>. - Загл. с экрана.
3. Электронная библиотека. Техническая литература [Электронный ресурс]. - Режим доступа: <http://techliter.ru/>. – Загл. с экрана.
4. Электронная библиотечная система Поволжского государственного университета сервиса [Электронный ресурс]. - Режим доступа: <http://elib.tolgas.ru/>. - Загл. с экрана.
5. 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	Операционная система Microsoft Windows	Системное ПО: операционная система Microsoft Windows	Выполнение лабораторных работ
2	Пакет Microsoft Office	Офисный пакет приложений. В состав этого пакета входит программное обеспечение для работы с различными типами документов: текстами, электронными таблицами, базами данных и др.	Выполнение лабораторных работ и оформление отчетов по лабораторным работам
4	Microsoft Visual Studio	В состав этого пакета входит программное обеспечение для создания приложений, на языках программирования Visual C++, Visual Basic, Visual JAVA.	Выполнение лабораторных работ
5	Ruby on Rails	Веб-ориентированная среда разработки Ruby on Rails.	Выполнение лабораторных работ
6	NetBeans	Интегрированная среда разработки NetBeans.	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 MS Office, ПО Microsoft Visual Studio, ПО Ruby on Rails, ПО NetBeans.

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.

