

Утверждено приказом ГОУ ВПО ДонГУУ от 23.08.2016г. №675

ДОНЕЦКАЯ НАРОДНАЯ РЕСПУБЛИКА
МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ
ГОСУДАРСТВЕННОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ
«ДОНЕЦКАЯ АКАДЕМИЯ УПРАВЛЕНИЯ И ГОСУДАРСТВЕННОЙ СЛУЖБЫ
ПРИ ГЛАВЕ ДОНЕЦКОЙ НАРОДНОЙ РЕСПУБЛИКИ»

ФАКУЛЬТЕТ ГОСУДАРСТВЕННОЙ СЛУЖБЫ И УПРАВЛЕНИЯ
КАФЕДРА ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ

УТВЕРЖДАЮ

Проректор по учебной работе

Л.Н.Костина

10.08.2017 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ

«Разработка программных приложений»

Направление подготовки 09.03.03 «Прикладная информатика»

Донецк
2017

2
Рабочая программа учебной дисциплины «Разработка программных приложений»
для студентов 4 курса образовательного уровня «бакалавр» направления подготовки
09.03.03 «Прикладная информатика» очной и заочной форм обучения.

Автор,

разработчик: старший преподаватель, Э.М. Лебезова

Программа рассмотрена на
заседании ПМК кафедры

«Прикладная информатика»

Протокол заседания ПМК от

08.06.2017

№ 10

Председатель ПМК



А. Н. Васильев

Программа рассмотрена на
заседании кафедры

Информационных технологий

Протокол заседания кафедры от

09.06.2017

№ 13

Заведующая кафедрой



Н. В. Братва

Рабочая программа учебной дисциплины «Разработка программных приложений» для студентов 4 курса образовательного уровня «бакалавр» направления подготовки 09.03.03 «Прикладная информатика» очной и заочной форм обучения.

Автор,

разработчик: старший преподаватель, Э.М. Лебезова

должность, ученая степень, ученое звание, инициалы и фамилия

Программа рассмотрена на
заседании ПМК кафедры

«Прикладная информатика»

Протокол заседания ПМК от

08.06.2017

№ 10

дата

Председатель ПМК

(подпись)

А. Н. Верзилов
(инициалы, фамилия)

Программа рассмотрена на
заседании кафедры

Информационных технологий

Протокол заседания кафедры от

09.06.2017

№ 13

дата

Заведующая кафедрой

(подпись)

Н. В. Брадул
(инициалы, фамилия)

1. Цель освоения дисциплины и планируемые результаты обучения по дисциплине (соотнесенные с планируемыми результатами освоения образовательной программы)

Цель изучения дисциплины – приобретение студентами теоретических знаний и практических навыков по программированию на языке Java и разработке кроссплатформенных клиентских и серверных приложений различного назначения.

Задачи учебной дисциплины:

- знакомство с технологиями разработки кроссплатформенных клиентских и серверных приложений различного назначения ;
- знакомство с сетевыми средствами Java;
- знакомство с объектной моделью Java;
- знакомство с основными библиотеками языка Java;
- знакомство с языком Java Script;
- приобретение практических навыков использования средств JEE для разработки web-приложений;
- приобретение практических навыков создания клиентских и серверных приложений различного назначения;
- приобретение практических навыков многопоточного программирования;
- приобретение практических навыков программирования апплетов и сервлетов.

Компетенции обучающегося, формируемые в результате освоения дисциплины:

Код соответствующей компетенции по ГОС	Наименование компетенций	Результат освоения (знать, уметь, владеть)
ОПК-1	способность использовать нормативно-правовые документы, международные и отечественные стандарты в области информационных систем и технологий	Знать: - нормативные правовые документы в области информационных систем и технологий. Уметь: - использовать нормативные правовые документы в области информационных систем и технологий для организации защиты информации. Владеть: - международными и отечественными стандартами в области разработки информационных систем и технологий.
ОПК-3	способность использовать основные законы естественнонаучных дисциплин и современные информационно-коммуникационные технологии в профессиональной деятельности	Знать: - парадигмы и методики разработки ПП. Уметь: - реализовывать эти подходы в конкретной предметной области. Владеть: - навыками работы с методами и типовыми средствами разработки ПП.

ОПК-4	способность решать стандартные задачи профессиональной деятельности на основе информационной и библиографической культуры с применением информационно-коммуникационных технологий и с учетом основных требований информационной безопасности	Знать: - информационные платформы различных разработчиков. Уметь: - применять методы анализа прикладной области на концептуальном, логическом, и алгоритмическом уровнях с целью выявления требований к информационной платформе. Владеть: - способностью применение этих методов для разработки ПП.
ПК-7	способность эксплуатировать и сопровождать ИС и сервисы	Знать: - принципы эксплуатации и сопровождения информационных систем и сервисов с точки зрения предметной области. Уметь: - эксплуатировать и сопровождать информационные системы и сервисы с точки зрения их эксплуатации. Владеть: - принципами эксплуатации и сопровождения информационных систем и сервисов с точки зрения их эффективного применения.
ПК-РПП	способность применять к решению прикладных задач базовые алгоритмы обработки информации, выполнять оценку сложности алгоритмов, программировать и тестировать программы	Знать: - как ставить и решать прикладные задачи с использованием современных информационно-коммуникационных технологий Уметь: - моделировать и проектировать структуры данных и знаний, прикладные и информационные процессы Владеть: - навыками создания ПП с использованием различных сред разработки

2. Место дисциплины в структуре основной образовательной программы

Дисциплина «Разработка программных приложений» относится к вариативной части профессионального цикла.

2.1. Требования к предварительной подготовке обучающегося

Дисциплина тесно связана с такими курсами, как: Базы данных, Информатика и программирование, Теория систем и системный анализ, Информационные системы и технологии, Проектирование информационных систем, Разработка информационных систем, Интернет-программирование.

2.2. Дисциплины и/или практики, для которых освоение данной дисциплины необходимо как предшествующее:

Основные положения дисциплины могут быть использованы в дальнейшем при

подготовке дипломной работы выпускниками, а также в дальнейшей практической деятельности.

3. Объем дисциплины в кредитах (зачетных единицах) с указанием количества академических часов, выделенных на аудиторную (по видам учебных занятий) и самостоятельную работу студента

	Кредиты ECTS (зачетные единицы)	Всего часов		Форма обучения	
		о	з	Очная	Заочная
				Семестр	
				№8	№8
Общая трудоемкость	3	108	108	Количество часов на вид работы:	
Виды учебной работы, из них:					
Аудиторные занятия (всего)		48	10	48	10
В том числе:					
Лекции					
Практические занятия		48	10	48	10
Самостоятельная работа (всего)		60	98	60	98
Промежуточная аттестация					
В том числе:					
экзамен				экзамен	экзамен

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

4.1. Разделы (темы) дисциплины с указанием отведенного на них количества академических часов и видов учебных занятий.

Наименование раздела, темы дисциплины	Виды учебной работы (бюджет времени)									
	Очная форма обучения					Заочная форма обучения				
	Лекции	Практические занятия	Семинарские занятия	Самостоятель ная работа	Всего	Лекции	Практические занятия	Семинарские занятия	Самостоятель ная работа	Всего
1	2	3	4	5	6	7	8	9	10	11
Раздел № 1. Объектно-ориентированное программирование на языке Java										
Тема 1.1. Управляющие конструкции и основные типы языка Java.		8		10	18		2		20	22
Тема 1.2. Обработка строк в java- приложениях. Регулярные выражения.		8		10	18		2		15	17
Тема 1.3. Структуры данных. Коллекции Java.		8		10	18		2		15	17

Наименование раздела, темы дисциплины	Виды учебной работы (бюджет времени)									
	Очная форма обучения					Заочная форма обучения				
	Лекции	Практические занятия	Семинарские занятия	Самостоятельная работа	Всего	Лекции	Практические занятия	Семинарские занятия	Самостоятельная работа	Всего
Итого по разделу:		24		30	54		6		50	54
Раздел № 2. Создание прикладных приложений НА ЯЗЫКЕ JAVA										
Тема 2.1. Разработка приложений с графическим интерфейсом пользователя.		8		10	18		2		15	17
Тема 2.2. Сетевые прикладные приложения		8		10	18		2		15	17
Тема 2.3. Разработка веб-приложений на Java. Сервлеты.		8		10	18				18	18
Итого по разделу:		24		30	54		4		50	54
Всего за семестр:		48		60	108		10		98	108

4.2. Содержание разделов дисциплины:

Наименование раздела, темы дисциплины	Содержание разделов дисциплины	Содержание семинарских/практических занятий		
		Кол-во часов		
		0	3	
1	2	3	4	5
Раздел 1. ОБЪЕКТНО-ОРИЕНТИРОВАННОЕ ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ JAVA				
Тема 1.1.	Введение в Java, обзор платформ и технологий. Управляющие конструкции языка Java. Простые типы данных Java. Массивы. Разработка консольных java-приложений.	Практическое занятие №1	8	2
		1. Среда разработки Java приложений Eclipse. 2. Использование массивов, строк и файлов.		
Тема 1.2.	Встроенные типы данных языка Java. Потoki ввода-вывода. Обработка строк Java. Использование регулярных выражений в java-приложениях. Классы и объекты. Класс Object. Передача и возврат объектов, дублирующие ссылки. Клонирование объектов.	Практическое занятие №2	8	2
		1. Обработка строк Java. 2. Использование регулярных выражений в java-приложениях.		
Тема 1.3.	Управление наборами данных в приложениях java. Специальные	Практическое занятие №3	8	2

Наименование раздела, темы дисциплины	Содержание разделов дисциплины	Содержание семинарских/практических занятий		
			Кол-во часов	
			0	3
1	2	3	4	5
	классы-коллекции, позволяющие создавать типовые структуры, такие как список, очередь, стек. Пакеты и интерфейсы, система обработки исключительных ситуаций.	1. Коллекции в java 2. Использование списков, очередей, стеков.		
Раздел № 2. Создание прикладных приложений НА ЯЗЫКЕ JAVA				
Тема 2.1.	Принципы построения графического интерфейса. Иерархия классов AWT и SWING. Графика в Java. Потoki и многопоточность, жизненный цикл потока.	Практическое занятие №4	8	2
		1. Графический интерфейс java-приложений.		
		2. Библиотеки Swing и AWT		
Тема 2.2.	Java и сеть. Аспекты разработки сетевых приложений, ориентированных на работу в сетях TCP/IP (как локальных, так и в сети Интернет). Сетевое программирование. Сокеты и сокетные соединения.	Практическое занятие №5	8	2
		1. Разработка простого сетевого клиента и сервера.		
		2. Создание веб-приложений. Апплеты.		
Тема 2.3.	Разработка веб-приложения (J2EE). Создание и синхронизация потоков. Сетевое программирование с сокетами и каналами.	Практическое занятие №6	8	
		Разработка простого сетевого клиента и сервера		
		Создание веб-приложений. Сервлеты		

5. Перечень учебной литературы, необходимой для освоения дисциплины

5.1. Перечень учебно-методического обеспечения для самостоятельной работы обучающихся по дисциплине

Элементы учебно-методического комплекса дисциплины утверждены на заседании кафедры информационных технологий (протокол №1 от 29.08.2017).

Все материалы представлены в онлайн-курсе по ссылкам:

<http://elearn.dsum.org/course/view.php?id=10> .

<http://elearn.dsum.org/course/view.php?id=18>

Контрольные вопросы для самоподготовки:

1. Дайте определения полиморфизма. Какие особенности его реализации на языке Java?
2. Назовите отличия индустриального и экстремального программирования. Какие приёмы программирования существуют для двух этих подходов в языке Java
3. Опишите основные отличия класса String от StringBuffer?
4. Что такое механизм автоматической сборки мусора (garbage collector)?
5. Основные классы библиотеки для разработки тестов JUNIT
6. Какие действия необходимо произвести, чтобы создать компонент и подписаться на событие, которое он генерирует?

7. Из каких частей состоит заголовок объявления класса? Тело класса?
8. Какие элементы языка Java имеют имена? Какие из них должны быть объявлены?
Какие элементы языка Java имеют имена? Какие из них должны быть объявлены?
9. Сравните использование операторов выбора switch в языке C# и Java.
10. Как отсортировать массив пользовательских объектов на языке Java ?
11. Когда для реализации списка типа List нужно использовать объект класса ArrayList, а когда LinkedList ?
12. Какие особенности использования коллекций для списков, множеств, пар значений?
13. Какие изменения для работы с коллекциями были внесены в версии Java 2 SE 5.0?
14. Ввести число, занести его цифры в стек. Вывести в число, у которого цифры идут в обратном порядке.
15. Задать два стека, поменять информацию местами.
16. Для каких элементов класса работает полиморфизм?
17. Нарисуйте иерархию классов ввода вывода на основе классов InputStream и OutputStream.
18. Какие модификаторы позволяют обращаться к элементу из классов того же пакета?
19. Как генерируется документация утилитой javadoc и набор воспринимаемой утилитой тегов в тексте программы для формирования документации, если в классе заводится новый элемент, и пока нет никаких факторов, позволяющих выбрать тот или иной модификатор доступа. Какой модификатор использовать в таком случае?
20. Есть ли какие-либо ограничения на набор интерфейсов, которые может реализовывать класс?
21. Нарисуйте иерархию классов ввода вывода на основе классов Reader и Writer ?
22. Как реализованы в классе Object методы equals(), toString(), hashCode()?
23. Какие объекты могут быть клонированы?
24. Какие условия должны быть выполнены при переопределении метода equals()?
25. Сколько объектов может быть создано в процессе выполнения клонирования одного объекта средствами JVM?
26. Хотя примитивные массивы не могут участвовать в преобразованиях, однако массивы int[][] и byte[][] могут рассматриваться как одномерные объектные массивы, основанные на ссылочном типе «одномерный примитивный массив». Могут ли такие типы быть преобразованы из одного в другой?
27. Как определить, можно ли преобразовать один тип массива к другому?
28. Сколько объектов порождается при инициализации массива new int[3][4]? new int[3][][]?
29. Для каких примитивных типов Java существуют классы-обертки? Что будет получено в результате выполнения: (new Integer(1)).equals(new Byte(1)) ?
30. Чем отличается внутренний (inner) класс от вложенного (nested) класса?
31. Может ли переменная иметь тип абстрактный класс? Интерфейс? Если да, то какие значения она может хранить?
32. Может ли переменная иметь тип абстрактный класс? Интерфейс? Если да, то какие значения она может хранить?
33. Как узнать, какие события генерирует стандартный графический компонент?
34. Опишите жизненный цикл потока.
35. Если объявить метод synchronized, то какой эффект будет этим достигнут?
36. Опишите механизмы синхронизации потоков
37. Как работают static synchronized методы?
38. Если один поток начал исполнение synchronized-блока, указав ссылку на некий объект, может ли другой поток обратиться к полю этого объекта? К методу?
39. Почему метод wait требует обработки InterruptedException, а методы notify и notifyAll – нет?
40. Может ли поток никогда не выйти из метода wait, даже если будет вызван метод notify? notifyAll?
41. Какие действия необходимо предпринять для установления TCP соединения между

42. Какие действия необходимо предпринять для установления TCP соединения между двумя Java-приложениями?
43. Какие действия необходимо предпринять для обмена данными по UDP протоколу?
44. Можно ли с помощью класса URL пересылать данные на сервер?
45. Для чего нужны менеджеры компоновки? Исходя из каких параметров они выполняют свою работу?
46. Напишите класс-компоненту, у которого по центру рисуется квадрат размерами 10x10.
47. Может ли быть дважды вызван метод `init` у апплета? Метод `start`?
48. Опишите на память иерархию классов, применяемых для построения визуальных приложений
49. Промоделировать вращение спутника вокруг планеты по эллиптической орбите. Когда скрывается за планетой – спутник не виден.
50. Создать фрейм и разместить на нем окружность (одну или несколько). Объект должен «убегать» от указателя мыши. При приближении на некоторое расстояние объект появляется в другом месте фрейма.
51. Имея два объекта класса `File`, каким образом будет наиболее корректно узнать, указывают ли они на одну и ту же директорию (и на директорию ли)? Возможно ли только с помощью этих двух объектов удалить директорию? Если да, то как изменится содержимое другого объекта (если они действительно указывают на одну и ту же директорию)?
52. Какие классы предоставляют методы для записи в поток двоичного представления значений примитивных типов Java?
53. Если необходимо записать (и после считать) несколько строк в файл (из файла), в каком порядке и какие следует настроить фильтры (и для чтения, и для записи)? Какие из них можно пропустить?
54. Что произойдет при попытке к одному объекту `PipedWriter` присоединить несколько различных объектов `PipedReader`? Что произойдет, если несколько раз подряд присоединять один и тот же `PipedReader`?
55. Какая кодировка используется классом `OutputStreamWriter` по умолчанию?
56. Что будет записано в поток, если вызвать метод `print()` класса `PrintWriter`, передав в качестве параметра `new File("d:\\word.txt")` ?
57. Какие значения могут быть переданы в конструктор `RandomAccessFile` для указания режима доступа (чтение/запись)?
58. Какое значение следует передать методу `seek()` объекта `RandomAccessFile`, чтобы последний байт файла был считан одиночным вызовом `read()`?
59. Какие методы объявлены в интерфейсе `Serializable`?
60. Что произойдет, если записать в файл, используя `ObjectOutputStream`, значения типов `long`, `int`, `byte` именно в таком порядке, а считать в обратном, используя `DataInputStream`?

5.2. Перечень основной учебной литературы

1. Гринберг А. С Информационный менеджмент: Учебное пособие [Электронный ресурс] / А. С. Гринберг. – Москва: ЮНИТИ-ДАНА, 2012. – 416с. – Режим доступа: <http://znanium.com/catalog.php?item=bookinfo&book=376745>
2. Джо Барнс. HTML5. Основы клиентской разработки: учебное пособие [Электронный ресурс] / Джо Барнс. – Москва: НОУ Интуит, 2016. – 161 с. – Режим доступа: <https://www.book.ru/book/917990>
3. Монахов, В.В. Язык программирования Java и среда NetBeans. 3-е изд., 2011 г. ЭБС Руконт
4. Машнин, Т.С. Web-сервисы Java, 2012г. ЭБС Руконт
5. Философия Java : научное издание / Б. Эккель ; Пер. с англ. Е. Матвеев. - 4-е изд. Санкт-Петербург : Питер, 2010. - 637 с.

6. Dan Woods, Thomas Mattern, "Enterprise SOA: Designing IT for Business Innovation", O'Reilly, 2006, ISBN: 0-596-10238-0, 452 p.
7. David Chappell, Tyler Jewell, Java Web Services, O'Reilly, 2002, 0-596-00269-6, 276 p.
8. Neil Gray, Web Server Programming, John Wiley & Sons Ltd, 2003, ISBN 0-470-85097-3, 621 p.
9. Norbert Bieberstein; Robert G. Laird; Dr. Keith Jones; Tilak Mitra. Executing SOA: A Practical Guide for the Service-Oriented Architect, IBM Press, 2008, ISBN: 0-13-235374-1, 240 p.
10. H. M. Deitel, P. J. Deitel. Java How to Program, Sixth Edition, Prentice Hall, 2004, ISBN-10 : 0-13-148398-6
11. Eric Pulier, Hugh Taylor. Understanding Enterprise SOA, Manning Publications Co., 2007, ISBN 1-932394-59-1, 242 p.
12. James McGovern, Sameer Tyagi, Michael Stevens and Sunil Matthew. Java Web Services Architecture, Morgan Kaufmann Publishers, 2003, ISBN:1558609008, 831 p
13. Пол Дж. Перроун, Венката Венката С. Р. "Кришна" Р. Чаганти Создание корпоративных систем на основе Java2 Enterprise Edition. Руководство разработчика. 2001 г. Издательство: "Вильямс" ISBN: 5845901685, 0672317958,
14. Марти Холл, Лэрри Браун. Программирование для Web. Библиотека профессионала, 2001 г. Издательство: "Вильямс"
15. Х. М. Дейтел, П. Дж. Дейтел, С. И. Сантри Технологии программирования на Java 2. Книга 1 Графика, JAVABEANS, Интерфейс пользователя Advanced Java 2 Platform. How to Program Издательство: Бином-Пресс, 2003 г. -560 с.: ил.
16. Х. М. Дейтел, П. Дж. Дейтел, С. И. Сантри Технологии программирования на Java 2. Книга 2 Распределенные приложения Advanced Java 2 Platform. How to Program Издательство: Бином-Пресс, 2003 г.-464с.: ил.
17. Х. М. Дейтел, П. Дж. Дейтел, С. И. Сантри Технологии программирования на Java 2. Книга 3. Корпоративные системы, сервлеты, JSP, Web-сервисы Advanced Java 2 Platform. How to Program Издательство: Бином-Пресс, 2003 г. -672 с.: ил.
18. Кришнамурти, Дж. Рексфорд Web-протоколы. Теория и практика. HTTP/1.1, взаимодействие протоколов, кэширование, измерение трафика Издательство: Бином, 2002 г.
19. Даконта М., Саганич А. XML и Java 2. Библиотека программиста, Издательство ПИТЕР, 2001 год, 384 с., ISBN 5-318-00187-4

5.3. Перечень дополнительной литературы

1. Блинов, И.Н., Романчик, В.С. Java 2. Практическое руководство. Мн.: УниверсалПресс, 2005. – 400с.
2. Буч Г. Объектно-ориентирование проектирование с примерами применения: Пер. с англ. – М.: Конкорд, 1992.
3. Кен Арнольд, Джеймс Гослинг - Язык программирования Java. С-Пб.: Изд-во Питер.– 1997 г.
4. Давыдов, С.В Профессиональное программирование на Java. – ВHV. 2005.
5. Дарвин, Я.Ф. Java: Сборник рецептов для профессионалов. -СПб.: Питер, 2002. – 764 с.
6. Эккель, Б. Философия Java. СПб.: Питер, 2001. – 880 с.
7. Хорстман, К.С., Корнелл, Г. Библиотека профессионала. Java 2. Основы. –Том 1. – М.: «Вильямс».– 2004. – 848 с.
8. Хорстман, К.С., Корнелл, Г. Библиотека профессионала. Java 2. Тонкости программирования. Том 2. – М.: «Вильямс», 2002. 1120 с.
9. Холмс, Д., Шилдт, Г. Искусство программирования на Java. –М.: «Вильямс», 2005. – 336 с.
10. Ноутон, П., Шилдт, Г. Java 2 в подлиннике. – ВHV, 2005. – 1072 с.
11. Фаулер М., Скотт К. UML в кратком изложении. Применение стандартного языка объектного моделирования: Пер. с англ. – М: Мир.: 1999.
12. Шилдт, Г. Java 2 v.5.0. [Электрон. ресурс] <http://bsu.iba.by/books/Java2v5.zip>

6. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», необходимых для освоения дисциплины

1. Java портал Sun Microsystems – <http://java.sun.com>.
2. Java-учебник [Электронный ресурс]. <http://www.java-study.ru/java-uchebnik>
3. Учебник по javascript [Электронный ресурс]. <http://www.learn.javascript.ru>

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

7.1. Перечень информационных технологий (при необходимости)

7.2. Перечень программного обеспечения (при необходимости)

При изучении дисциплины используется ПО в составе: IDE Eclipse, IDE Ideas, сервер баз данных MySQL с управлением через phpMyadmin.

7.3. Перечень информационных справочных систем (при необходимости)

В процессе изучения дисциплины используются возможности информационно-справочной системы портала <http://window.edu.ru/>.

Сайт с учебными курсами по различным аспектам создания электронных порталов: <http://www.intuit.ru>

8. Фонд оценочных средств для контроля уровня сформированности компетенций

8.1. Виды промежуточной аттестации.

Текущий контроль успеваемости позволяет оценить уровень сформированности элементов компетенций (знаний и умений), компетенций с последующим объединением оценок и проводится в форме устного опроса (фронтальный, индивидуальный, комплексный), письменной проверки (ответы на вопросы, тестовые задания), включая задания для самостоятельной работы.

Основными формами текущего контроля знаний являются:

- решение прикладных задач средствами информационных технологий, уяснение эффективных подходов к выбору инструментальных средств и их применению в различных проблемных ситуациях;
- проверка качества усвоения проблемных вопросов изучаемого материала в ходе плановых занятий, обсуждение вынесенных в планы практических занятий контрольных вопросов;
- выполнение и защита плановой практической работы;
- компьютерное тематическое тестирование по теоретическим вопросам дисциплины, использование электронных тренажеров и комплексов для решения задач и проверки практических навыков студентов.

Промежуточная аттестация в форме экзамена позволяет оценить уровень сформированности компетенций в целом по дисциплине осуществляется в устной форме.

8.2. Показатели и критерии оценки результатов освоения дисциплины.

Средним баллом за дисциплину является средний балл за текущую учебную деятельность.

Механизм конвертации результатов изучения студентом дисциплины в оценки по традиционной (государственной) шкале и шкале ECTS представлен в таблице.

Средний балл по дисциплине (текущая успеваемость)	Отношение полученного студентом среднего балла по дисциплине к максимально возможной величине этого показателя	Оценка по государственной шкале	Оценка по шкале ECTS	Определение
4,5 – 5,0	90% – 100%	5	A	отлично – отличное выполнение с незначительным количеством неточностей (до 10%)
4,0 – 4,49	80% – 89%	4	B	хорошо – в целом правильно выполненная работа с незначительным количеством ошибок (до 20%)
3,75 – 3,99	75% – 79%	4	C	хорошо – в целом правильно выполненная работа с незначительным количеством ошибок (до 25%)
3,25 – 3,74	65% – 74%	3	D	удовлетворительно – неплохо, но со значительным количеством недостатков (до 35%)
3,0 – 3,24	60% – 64%	3	E	достаточно – выполнение удовлетворяет минимальные критерии, но со значительным количеством недостатков (до 40%)
до 3,0	35% – 59%	2	FX	неудовлетворительно с возможностью повторной сдачи (ошибок свыше 40%)
	0 – 34%	2	F	неудовлетворительно – надо поработать над тем, как получить положительную оценку (ошибок свыше 65%)

8.3. Критерии оценки работы студента.

При усвоении каждой темы за текущую учебную деятельность студента выставляются оценки по 5-балльной (государственной) шкале. Оценка за каждое задание в процессе текущей учебной деятельности определяется на основе процентного отношения операций, правильно выполненных студентом во время выполнения задания:

- 90-100% – «5»,
- 75-89% – «4»,
- 60-74% – «3»,
- менее 60% – «2».

Если на занятии студент выполняет несколько заданий, оценка за каждое задание выставляется отдельно.

8.3.1. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или опыта деятельности, характеризующие этапы формирования компетенций в процессе освоения образовательной программы)

Контрольные задания

Раздел № 1. Объектно-ориентированное программирование на языке Java.

Тема 1.1. Управляющие конструкции и основные типы языка Java.

1. Ввести n строк с консоли, найти самую короткую строку. Вывести эту строку и ее длину.
2. Ввести n строк с консоли. Упорядочить и вывести строки в порядке возрастания их длин, а также (второй приоритет) значений этих их длин.
3. Ввести n строк с консоли. Вывести на консоль те строки, длина которых меньше средней, также их длины.
4. В каждом слове текста k -ю букву заменить заданным символом. Если k больше длины слова, корректировку не выполнять.
5. В русском тексте каждую букву заменить ее номером в алфавите. В одной строке печатать текст с двумя пробелами между буквами, в следующей строке внизу под каждой буквой печатать ее номер.
6. Из небольшого текста удалить все символы, кроме пробелов, не являющиеся буквами. Между последовательностями подряд идущих букв оставить хотя бы один пробел.
7. Из текста удалить все слова заданной длины, начинающиеся на согласную букву.
8. В тексте найти все пары слов, из которых одно является обращением другого.
9. Найти и напечатать, сколько раз повторяется в тексте каждое слово.
10. Найти, каких букв, гласных или согласных, больше в каждом предложении текста.
11. Выбрать три разные точки заданного на плоскости множества точек, составляющие треугольник наибольшего периметра.
12. Найти такую точку заданного на плоскости множества точек, сумма расстояний от которой до остальных минимальна.
13. Выпуклый многоугольник задан на плоскости перечислением координат вершин в порядке обхода его границы. Определить площадь многоугольника.

Раздел № 1. Объектно-ориентированное программирование на языке Java.

Тема 1.2. Обработка строк в java-приложениях. Регулярные выражения.

1. Написать регулярное выражение, определяющее является ли данная строка строкой "abcdefghijklmnpqrstuv18340" или нет.
 - пример правильных выражений: abcdefghijklmnpqrstuv18340.
 - пример неправильных выражений: abcdefghijklmnoasdfasdpqrstuv18340.
2. Написать регулярное выражение, определяющее является ли данная строка GUID с или без скобок. Где GUID это строчка, состоящая из 8, 4, 4, 4, 12

- шестнадцатеричных цифр разделенных тире.
 – пример правильных выражений: e02fd0e4-00fd-090A-ca30-0d00a0038ba0.
 – пример неправильных выражений: e02fd0e400fd090Aca300d00a0038ba0.
3. Написать регулярное выражение, определяющее является ли заданная строка правильным MAC-адресом.
 – пример правильных выражений: aE:dC:cA:56:76:54.
 – пример неправильных выражений: 01:23:45:67:89:Az.
 4. Написать регулярное выражение, определяющее является ли данная строка валидным URL адресом. В данной задаче правильным URL считаются адреса http и https, явное указание протокола также может отсутствовать. Учитываются только адреса, состоящие из символов, т.е. IP адреса в качестве URL не присутствуют при проверке. Допускаются поддомены, указание порта доступа через двоеточие, GET запросы с передачей параметров, доступ к подпапкам на домене, допускается наличие якоря через решетку. Однобуквенные домены считаются запрещенными. Запрещены спецсимволы, например «-» в начале и конце имени домена. Запрещен символ «_» и пробел в имени домена. При составлении регулярного выражения ориентируйтесь на список правильных и неправильных выражений заданных ниже.
 – пример правильных выражений: http://www.example.com, http://example.com.
 – пример неправильных выражений: Just Text, http://a.com.
 5. Написать регулярное выражение, определяющее является ли данная строка шестнадцатеричным идентификатором цвета в HTML. Где #FFFFFF для белого, #000000 для черного, #FF0000 для красного и т.д.
 – пример правильных выражений: #FFFFFF, #FF3421, #00ff00.
 – пример неправильных выражений: 232323, #fddee, #fd2.
 6. Написать регулярное выражение, определяющее является ли данная строка датой в формате dd/mm/уууу. Начиная с 1600 года до 9999 года.
 – пример правильных выражений: 29/02/2000, 30/04/2003, 01/01/2003.
 – пример неправильных выражений: 29/02/2001, 30-04-2003, 1/1/1899.
 7. Написать регулярное выражение, определяющее является ли данная строка валидным E-mail адресом согласно RFC под номером 2822.
 – пример правильных выражений: user@example.com, root@localhost
 – пример неправильных выражений: bug@.@@com.ru, @val.ru, Just Text2.
 8. Составить регулярное выражение, определяющее является ли заданная строка IP адресом, записанным в десятичном виде.
 – пример правильных выражений: 127.0.0.1, 255.255.255.0.
 – пример неправильных выражений: 1300.6.7.8, abc.def.gha.bcd.
 9. Проверить, надежно ли составлен пароль. Пароль считается надежным, если он состоит из 8 или более символов. Где символом может быть английская буква, цифра и знак подчеркивания. Пароль должен содержать хотя бы одну заглавную букву, одну маленькую букву и одну цифру.
 – пример правильных выражений: C00l_Pass, SupperPas1.
 – пример неправильных выражений: Cool_pass, C00l.
 10. Проверить является ли заданная строка шестизначным числом, записанным в десятичной системе счисления без нулей в старших разрядах.
 – пример правильных выражений: 123456, 234567.
 – пример неправильных выражений: 1234567, 12345.
 11. Есть текст со списками цен. Извлечь из него цены в USD, RUR, EU.
 – пример правильных выражений: 23.78 USD.
 – пример неправильных выражений: 22 UDD, 0.002 USD.
 12. Проверить существуют ли в тексте цифры, за которыми не стоит «+».
 – пример правильных выражений: (3 + 5) – 9 × 4.
 – пример неправильных выражений: 2 * 9 – 6 × 5.
 13. Создать запрос для вывода только правильно написанных выражений со скобками (количество открытых и закрытых скобок должно быть одинаково).

– пример правильных выражений: $(3 + 5) - 9 \times 4$.

– пример неправильных выражений: $((3 + 5) - 9) \times 4$.

Раздел № 1. Объектно-ориентированное программирование на языке Java.

Тема 1.3. Структуры данных. Коллекции Java.

1. Ввести строки из файла, записать их в стек. Вывести строки в файл в обратном порядке.
2. Ввести число, занести его цифры в стек. Вывести в число, у которого цифры идут в обратном порядке.
3. Сложить два многочлена заданной степени, если коэффициенты многочленов хранятся в объекте HashMap.
4. Создать стек из элементов каталога.
5. Не используя вспомогательных объектов, переставить отрицательные элементы данного списка в конец, а положительные - в начало этого списка.
6. Организовать вычисления в виде стека.
7. Выполнить попарное суммирование произвольного конечного ряда чисел следующим образом: на первом этапе суммируются попарно рядом стоящие числа, на втором этапе суммируются результаты первого этапа и т.д. до тех пор, пока не останется одно число.
8. Задать два стека, поменять информацию местами.
9. Определить класс Stack. Объявить объект класса. Ввести последовательность символов и вывести ее в обратном порядке.
10. Умножить два многочлена заданной степени, если коэффициенты многочленов хранятся в списках.
11. Определить класс Set на основе множества целых чисел, $n =$ размер. Создать методы для определения пересечения и объединения множеств.
12. Программа получает N параметров вызова (аргументы командной строки). Эти параметры – элементы вектора. Строится массив типа double, а на базе этого массива – объект класса DoubleVector. Далее программа выводит в консоль значения элементов вектора в виде: Вектор: 2.3 5.0 7.3.
13. Списки (стеки) $I(1..N)$ и $U(1..N)$ содержат результаты N измерений тока и напряжения на неизвестном сопротивлении R . Найти приближённое число R методом наименьших квадратов.

Раздел № 2. Создание прикладных приложений на языке JAVA

Тема 2.1. Разработка приложений с графическим интерфейсом пользователя

1. Задать движение по экрану строк (одна за другой) из массива строк. Направление движения по апплету и значение каждой строки выбирается случайным образом.
2. Задать движение окружности по апплету так, чтобы при касании границы окружность отражалась от нее с эффектом упругого сжатия.
3. Изобразить в апплете приближающийся издали шар, удаляющийся шар. Шар должен двигаться с постоянной скоростью.
4. Изобразить в окне приложения отрезок, вращающийся в плоскости экрана вокруг одной из своих концевых точек. Цвет прямой должен изменяться при переходе от одного положения к другому.
5. Изобразить в окне приложения отрезок, вращающийся в плоскости фрейма вокруг точки, движущейся по отрезку.
6. Изобразить четырехугольник, вращающийся в плоскости апплета вокруг своего центра тяжести.
7. Создать фрейм с областью для рисования «пером». Создать меню для выбора цвета и толщины линии.

8. Составить программу для управления скоростью движения точки по апплету. Одна кнопка увеличивает скорость, другая – уменьшает. Каждый щелчок изменяет скорость на определенную величину.
9. Изобразить в окне гармонические колебания точки вдоль некоторого горизонтального отрезка. Если длина отрезка равна q , то расстояние от точки до левого конца в момент времени t можно считать равным $q(1 + \cos(\omega t))/2$, где ω – некоторая константа. Предусмотреть поля для ввода указанных величин и кнопку для остановки и пуска процесса.
10. Создать апплет со строкой движущейся по диагонали. При достижении границ апплета все символы строки случайным образом меняют регистр. При этом шрифт меняется на шрифт, выбранный из списка.
11. Создать апплет со строкой движущейся горизонтально, отражаясь от границ апплета и меняя при этом свой цвет, на цвет выбранный из выпадающего списка.
12. Промоделировать вращение спутника вокруг планеты по эллиптической орбите. Когда скрывается за планетой – спутник не виден.
13. Промоделировать аналоговые часы (со стрелками) с кнопками для увеличения/уменьшения времени на час/минуту.

Раздел № 2. Создание прикладных приложений на языке JAVA

Тема 2.2. Сетевые прикладные приложения

Создать на основе сокетов клиент/серверное визуальное приложение

1. Клиент посылает через сервер сообщение другому клиенту.
2. Клиент посылает через сервер сообщение другому клиенту, выбранному из списка.
3. Чат. Клиент посылает через сервер сообщение, которое получают все клиенты. Список клиентов хранится на сервере в файле.
4. Клиент при обращении к серверу получает случайно выбранный сонет Шекспира из файла.
5. Сервер рассылает сообщения выбранным из списка клиентам. Список хранится в файле.
6. Сервер рассылает сообщения в определенное время определенным клиентам.
7. Сервер рассылает сообщения только тем клиентам, которые в настоящий момент находятся в on-line.
8. Чат. Сервер рассылает всем клиентам информацию о клиентах вошедших в чат и покинувших его.
9. Клиент выбирает изображение из списка и пересылает его другому клиенту через сервер.
10. Игра по сети в «Морской бой».
11. Игра по сети в «21».
12. Игра по сети в классические «Крестики-нолики».
13. Игра по сети в «Го» («крестики-нолики» на безразмерном (большом) поле. Для победы необходимо выстроить пять в один ряд).

Раздел № 2. Создание прикладных приложений на языке JAVA

Тема 2.3. Разработка веб-приложений на Java. Сервлеты.

Создать сервлет и взаимодействующие с ним пакеты Java-классов и HTML-документов.

1. Генерация таблиц по переданным параметрам: заголовок, количество строк и столбцов, цвет фона.
2. Вычисление тригонометрических функций в градусах и радианах с указанной точностью. Выбор функций должен осуществляться через выпадающий список.
3. Поиск слова, введенного пользователем. Поиск и определение частоты встречаемости осуществляется в текстовом файле, расположенном на сервере.
4. Вычисление объемов тел (параллелепипед, куб, сфера, тетраэдр, тор, шар, эллипсоид и т.д.) с точностью и параметрами, указываемыми пользователем.
5. Поиск и (или) замена информации в коллекции по ключу (значению).

6. Выбор текстового файла из архива файлов по разделам (поэзия, проза, фантастика и т.д.) и его отображение.
7. Выбор изображения по тематике (природа, автомобили, дети и т.д.) и его отображение.
8. Информация о среднесуточной температуре воздуха за месяц задана в виде списка, хранящегося в файле. Определить: а) среднемесячную температуру воздуха; б) количество дней, когда температура была выше среднемесячной; в) количество дней, когда температура опускалась ниже; г) три самых теплых дня.
9. Реализация адаптивного теста из цепочки в 3 – 4 вопроса.
10. Вывод фрагментов текстов шрифтами различного размера. Размер шрифта и количество строк задается на стороне клиента.
11. Информация о точках на плоскости хранится в файле. Выбрать все точки, наиболее приближенные к заданной прямой. Параметры прямой и максимальное расстояние от точки до прямой вводятся на стороне клиента.
12. Осуществить сортировку введенного пользователем массива целых чисел. Числа вводятся через запятую.
13. Осуществить форматирование выбранного пользователем текстового файла, так чтобы все абзацы имели отступ ровно 3 пробела, а длина каждой строки была ровно 80 символов и не имела начальными и конечными символами пробел.

Критерии оценивания компетенций (результатов) по уровням освоения учебного материала:

1 – репродуктивный (освоение знаний, выполнение деятельности по образцу, инструкции или под руководством), если самостоятельно (или с помощью преподавателя) выполнены все пункты работы;

2 – продуктивный (планирование и самостоятельное выполнение деятельности, решение проблемных задач; применение умений в новых условиях), если выполнены все пункты работы самостоятельно и улучшена точность результата;

3 – творческий (самостоятельное проектирование экспериментальной деятельности; оценка и самооценка инновационной деятельности), если предложен более рациональный алгоритм решения задачи.

8.3.2. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и/или опыта деятельности

– оценивание проводится преподавателем в течении всего учебного процесса на основе выполнения текущих контрольных и индивидуальных заданий, самостоятельной работы за компьютером;

– результаты выполнения практических работ предъявляются в виде отчетов в электронном виде;

– оценивание практических работ осуществляет преподаватель, который проводит практические занятия.

– экзамен принимает преподаватель, который проводит практические занятия.

9. Методические указания для обучающихся по освоению дисциплины

Методические указания по дисциплине разработаны для всех практических работ. Вместе с индивидуальными заданиями по каждой практической работе и вопросами для самостоятельной работы они составляют методический комплект, доступный студентам в электронном виде.

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

При изучении дисциплины используются компьютерные классы первого корпуса (704 или 705). Компьютерные классы оборудованы ПК IBM PC (Core I5-750 2.66 GHz, MB S-1156 ASUS P7P55 Lx, DDR3 2200 4 Gb, PCI Express, HDD 1,5 Tb), лекционные

аудитории – мультимедийной проекционной техникой в стационарном или переносном варианте. Чтение лекций сопровождается демонстраций электронных презентаций по каждой теме.

11.Иные сведения и (или) материалы: (включаются на основании решения кафедры)

