Министерство образования и науки Российской Федерации
Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
**«Владимирский государственный университет
имени Александра Григорьевича и Николая Григорьевича Столетовых»
(ВлГУ)**

УТВЕРЖДАЮ
Проректор по УМР
_____ А.А.Панфилов
«17» 04 2015 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

Разработка кросс-платформенных приложений на Java

(наименование дисциплины)

Направление подготовки: 02.03.03 «Математическое обеспечение и администрирование информационных систем»

Профиль/программа подготовки

Уровень высшего образования: бакалавриат

Форма обучения: очно-заочная

Семестр	Трудоемкость зач. ед,/ час.	Лекции, час.	Практич. занятия, час.	Лаборат. работы, час.	СРС, час.	Форма промежуточного контроля (экз./зачет)
5	4/144	18	-	36	90	Зачет с оценкой
Итого	4/144	18	-	36	90	Зачет с оценкой

Владимир 2015

1. ЦЕЛИ ОСВОЕНИЯ ДИСЦИПЛИНЫ

Целью данного курса является изучение объектно-ориентированного языка программирования Java и основных приемов разработки кросс-платфоменных приложений на платформе Java 2 Standart Edition (J2SE).

Кроме того, в процессе освоения данной дисциплины у студента формируется понимание сути и значимости концепции проектирования Model-View-Controller (MVC) при разработке архитектуры приложений.

2. МЕСТО ДИСЦИПЛИНЫ В СТРУКТУРЕ ОПОП ВО

Дисциплина относиться к дисциплинам по выбору вариативной части ОПОП. Изучение данной дисциплины проходит в 5-м семестре и базируется на знаниях, приобретённых студентами в рамках общеобразовательных курсов по программированию:

- "Основы программирования"
- "Объектно-ориентированное программирование"

Для усвоения курса необходимо:

- знание основ процедурного программирования
- знание основ объектно-ориентированного программирования
- умение самостоятельно разрабатывать и тестировать приложения на одном из языков программирования высокого уровня (Pascal, C, C++, C#).

Для успешного усвоения курса приветствуется знание языка C++ и глубокое понимание его объектно-ориентированных возможностей.

Знания и практические навыки данного курса могут быть применены:

- при написании курсовых работ по смежным дисциплинам, требующим знания языков и технологий программирования
- при написании выпускной квалификационной работы
- для профессионального использования при трудоустройстве в IT-компаниии, занимающиеся разработкой программного обеспечения на платформе Java 2 Standart Edition (J2SE)

3. КОМПЕТЕНЦИИ ОБУЧАЮЩЕГОСЯ, ФОРМИРУЕМЫЕ В РЕЗУЛЬТАТЕ ОСВОЕНИЯ ДИСЦИПЛИНЫ (МОДУЛЯ)

В результате освоения дисциплины у студента должны быть сформированы общепрофессиональные и профессиональные компетенции, указанные в учебном плане, так же студент должен демонстрировать следующие результаты обучения:

1. **Знать:** кроссплатформенный объектно-ориентированный язык программирования Java; основные пакеты платформы Java 2 Standart Edition (J2SE); концепцию проектирования Model-View-Controller (MVC) (компетенция ОПК-4: способность применять в профессиональной деятельности основные методы и средства автоматизации проектирования, производства, испытаний и оценки качества программного; компетенция ОПК-8: способность использовать знания методов проектирования и производства программного продукта, принципов построения, структуры и приемов работы с инструментальными средствами, поддерживающими создание программного обеспечения)

2. **Владеть:** навыками анализа исходной задачи, проектирования архитектуры приложения и реализации программного кода (компетенция ОПК-3: готовность анализировать проблемы и направления развития технологий программирования; компетенция ОПК-11: готовность использовать навыки выбора, проектирования, реализации, оценки качества и анализа эффективности программного обеспечения для решения задач в различных предметных областях; ОПК-6: способностью определять проблемы и тенденции развития рынка программного обеспечения).

Уметь: разрабатывать клиент-серверные приложения с многопоточной архитектурой и оконным пользовательским интерфейсом (компетенция ОПК-9: способность исполь-

зовать знания методов организации работы в коллективах разработчиков ПО, направления развития методов и программных средств коллективной разработки ПО).

1. СТРУКТУРА И СОДЕРЖАНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Общая трудоемкость дисциплины составляет 4 зачетных единц, 144 часа.

№ п/п	Раздел (тема) дисциплины	Семестр	Неделя семестра	Виды учебной работы, включая самостоятельную работу студентов и трудоемкость (в часах)						Объем учебной работы, с применением интерактивных методов (в часах / %)	Формы текущего контроля успеваемости *(по неделям семестра)*, форма промежуточной аттестации *(по семестрам)*
				Лекции	Практические занятия	Лабораторные работы	Контрольные работы	СРС	КП / КР		
1	Введение в дисциплину	5	1	1	-	2	-	-		1,5/50%	Рейтинг-контроль 1
2	Основы языка Java	5	1-2	1	-	-	-	8	-	0,5/50%	
3	Объектная модель Java	5	3-4	2	-	4	-	4	-	3/50%	
4	Массивы примитивных и ссылочных типов	5	5-6	2	-	-	-	4	-	2/50%	
5	Приведение типов	5	7	2	-	-	-	4	-	1/50%	
6	Пакет java.util: коллекции	5	8	-	-	6	-	4	-	3/50%	Рейтинг-контроль 2
7	Исключения	5	9	2	-	4	-	4	-	4/50%	
8	Потоки данных(stream), пакет java.io	5	10-11	2	-	4	-	4	-	3/50%	
9	Работа с сетью, пакет java.net	5	11-12	2	-	4	-	4	-	3/50%	
10	Потоки выполнения	5	12-13	2	-	4		4		3/50%	

- абстрактные классы;
- интерфейсы;
- механизм позднего связывания и полиморфизм
- класс java.lang.Object;
- класс java.lang.Class

Массивы примитивных и ссылочных типов

Приведение типов:

- приведение примитивных и ссылочных типов;
- запрещенные приведения

Пакет java.util: коллекции

Исключения:

- понятие исключительной ситуации (ИС)
- причины возникновения ИС;
- классификация ИС;
- обработка ИС (конструкция try-catch и try-catch-finally);
- оператор throw;
- пользовательские классы исключений

Потоки данных(stream), пакет java.io:

- система ввода/вывода,
- сериализация;
- классы java.io.Reader и java.io.Writer;
- класс java.io.File

Работа с сетью, пакет java.net:

- сетевые протоколы;
- классы java.net.InetAdress, java.net.Socket и java.net.ServerSocket

Потоки выполнения:

- касс java.lang.Thread;
- интерфейс java.lang.Runnable;
- демон-потоки;
- синхронизация;
- методы wait(), notify(), notifyAll() класса Object;
- могопоточная архитектура в клиент-серверных приложениях

Пользовательский интерфейс, пакет java.awt:

- дерево компонент;
- принципы отрисовки;
- модель сообщений;
- менеджеры компоновки

Архитектурный шаблон проектирования MVC на примере тестовой задачи

Разработка клиент-серверных приложений с многопоточной архитектурой и пользовательским интерфейсом:

- анализ предметной области;
- use case diagram, class diagram, activity diagram;
- разработка архитектуры приложения в концепции MVC
- проектирование пользовательского интерфейса на базе AWT;
- программная реализация

Практическая часть курса состоит из лабораторных работ и курсовой работы.

5. ОБРАЗОВАТЕЛЬНЫЕ ТЕХНОЛОГИИ

- лекционно-семинарская система обучения (традиционные лекционные и лабораторные занятия);
- case-study (получение на лабораторных работах учебных кейсов с постановкой задачи и методической проработкой изучаемой темы);

- обучение в малых группах (выполнение лабораторных работ в группах из двух или трёх человек);
- применение мультимедиа технологий (проведение лекционных и семинарских занятий с применением компьютерных презентаций и демонстрационных роликов с помощью проектора или ЭВМ);
- технология развития критического мышления (прививание студентам навыков критической оценки полученных решений)

6. ОЦЕНОЧНЫЕ СРЕДСТВА ДЛЯ ТЕКУЩЕГО КОНТРОЛЯ УСПЕВАЕМОСТИ, ПРОМЕЖУТОЧНОЙ АТТЕСТАЦИИ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ И УЧЕБНО-МЕТОДИЧЕСКОЕ ОБЕСПЕЧЕНИЕ САМОСТОЯТЕЛЬНОЙ РАБОТЫ СТУДЕНТОВ[1]

ТЕКУЩИЙ КОНТРОЛЬ

Текущим контролем успеваемости является действующая в университете система рейтинг-контроля.

Ниже приводится примерные вопросы первого и второго рейтинг-контроля по разделам.

Вопросы рейтинг-контроля №1(примерные вопросы):

Раздел №1

1. Понятие кроссплатфрменного программного обеспечения
2. Платформа J2SE: основные пакеты
3. Из какой директории необходимо запускать компилятор, чтобы скомпилировать Java-приложение, состоящее из одного класса: test.first.Start, описание которого сохранено в файле Start.java по маршруту - c:\Java\programs\test\first\?
4. Из какой директории необходимо запускать интерпретатор Java для выполнения приложения, состоящее из одного класса: test.first.Start, описание которого сохранено в файле Start.java по маршруту - c:\Java\programs\test\first\?

Раздел №2

5. Выберите верное утверждение - в Java определены следующие лексемы:
   1) идентификаторы, ключевые слова, литералы, разделители, операторы
   2) ключевые слова, литералы, разделители
   3) ключевые слова, литералы, операторы
   4) ключевые слова, литералы
   5) идентификаторы, ключевые слова, операторы
   6) нет верных вариантов
6. Перечислите все виды литералов в Java
7. Какие из перечисленных идентификаторов являются корректными?
   1) abc
   2) 1ab
   3) _bc
   4) _1c $ac
   5) $ac
   6) for_
   7) Int
   8) Byte
8. Что такое комментарий разработчика и для чего он служит?

---

[1] При разработке оценочных средств были использованы материалы из источников, указанных в разделе 7 Рабочей программы

9. Являются ли следующие слова ключевыми:
   1) true
   2) goto
   3) null
   4) const
   5) false
10. Чему будет равно следующее выражение и значения переменных x и y после вычислений?

```
int x=0, y=0;
print((++x==1) || (y++==1));
```

11. Что будет в результате работы программного кода:

```
byte b=3;
int c=b;
c++;
print(++b==c);
```

12. Что будет в результате выполнения выражений:
   1) 1/2
   2) 1./2
   3) 1/2.
   4) 1./2.
13. Выберите верные утверждения:
   1) 123 является целочисленным литералом
   2) -5 является целочисленным литералом
   3) целочисленные литералы имеют тип int по умолчанию
   4) инициализация переменных числовых типов литералами, выходящими за допустимый диапазон значений, даст ошибку времени выполнения
   5) инициализация переменных числовых типов литералами, выходящими за допустимый диапазон значений, на этапе выполнения будет выполнена с потерей точности (произойдет отбрасывание старших бит, искажение знакового разряда, округление до последнего допустимого разряда и т.п.)
   6) нет верных утверждений
14. Проанализировать программный код на предмет синтаксической и семантической корректности - даст ли код ошибки компиляции и ошибки времени выполнения. Если код корректен, то что будет выведено на экран. Ответ пояснить

```
1) float a = 1.0f / 0.0f;
   System.out.println("a = " + a);
2) float b = 0.0f / 0.0f;
   System.out.println("b = " + b);
3) float c = (1.0f / 0.0f) * 0.0f;
   System.out.println("c = " + c);
4) float d = 1.0f / +0.0f;
   System.out.println("d = " + d);
5) float e = 5.5;
   System.out.println("e = " + e);
6) boolean f = 1;
   System.out.println("f = " + f);
```

15. Корректно ли следующее объявление с точки зрения формального выполнения соглашений по именованию:

```
public class flat{
        private int floor_number;
        private int r; // количество комнат
        public int rooms() {
```

```
            retun r;
        }
        public int GetFloorNumber() {
        return floor_number;
        }
    }
```

16. Могут ли пакет и вложенный пакеты содержать одноименные классы?

Раздел №3

17. Объявление класса: сигнатура и тело класса
18. Полная сигнатура методов класса
19. Если метод использует поле класса, должно ли оно быть объявлено выше объявления метода?
20. Что такое конструктор? Какие бывают конструкторы?
21. Может ли класс не иметь ни одного конструктора?
22. Что такое this?
23. Что такое super?
24. Можно ли при наследовании не реализовывать абстрактный метод родительского класса?
25. Если есть ссылка на абстрактный класс, можно ли с ее помощью обращаться к абстрактным методам этого класса?
26. Возможно ли не реализовывать все методы из интерфейса, указанного в выражении implements?
27. Какие модификаторы позволяют обращаться к элементу из классов того же пакета?
28. Какие модификаторы элементов интерфейса подставляются по умолчанию, а потому не рекомендованы для явного указания?
29. Выберите верные утверждения:
    1) у пакетов нет модификаторов доступа, любой пакет может использоваться из любой точки программы
    2) объявление локального класса с явным указанием модификатора доступа вызовет ошибку компиляции
    3) все элементы интерфейсов являются public.
    4) конструкторы не наследуются, поэтому не могут иметь модификатор доступа protected
    5) прямое обращение к закрытому элементу класса из методов других классов вызовет ошибку компиляции
    6) прямое обращение к закрытому элементу класса из методов других классов не вызовет ошибку компиляции, но даст исключение SecurityException на этапе выполнения
    7) нет верных утверждений
30. При объявлении классов и интерфейсов существует возможность указать:
    1) модификатор public или не указывать его (уровень доступа назначается по умолчанию)
    2) модификаторы public, protected или не указывать никакой (уровень доступа назначается по умолчанию)
    3) модификатор public, protected, private или не указывать никакой (уровень доступа назначается по умолчанию)
    4) классы и интерфейсы не имеют модификаторов доступа, любой класс или интерфейс может использоваться из любой точки программы.
    5) нет верных утверждений
31. Для каких элементов класса работает полиморфизм?
32. Какое значение появится на консоли после выполнения следующего программного кода:

```
public class Parent {
        int x = 2;
        public void print() {
                System.out.println(x);
        }
}
public class Child extends Parent {
        int x = 3;
        public static void main(String s[]) {
                new Child().print();
        }
}
```

33. Эквиваленты ли две следующие операции над ссылочными переменными x1 и x2, если SomeClass2 – это тип переменной x2:
    x1 instanceof SomeClass2
    x1.getClass().getName().equals(x2.getClass().getName());
34. При каком значении ссылочной переменной x следующее выражение всегда будет возвращать истину:
    x.getClass()==x;
35. Как реализованы в классе Object методы equals(), toString()?
36. При переопределении метода equals(), будет ли требоваться переопределение каких-либо других методов, для корректного использования объектов рассматриваемого класса в хэш-таблицах?

Раздел №4

37. От какого класса наследуются классы массивов?
38. Какая размерность у следующих массивов:
    1) int x[], y[][];
    2) String s, s1[], s2={{}, {"a", "b"}, null};
39. Можно ли использовать инициализатор в выражении выделения памяти под элементы массива? Если да, то дополнить программный код инициализатором:
    int x[][]=new int[2][3];
40. Сколько объектов порождается при инициализации массивов:
    1) int a[][] = new int[3][4];
    2) int b[][] = new int[3][][];
41. Что будет в результате выполнения следующего программного кода:
```
Point p2[][] = (Point[][])p1.clone();
p2[0]= new Point[]{new Point(2, 2)};
System.out.println(p1[0][0]);
```

Вопросы рейтинг-контроля №2(примерные вопросы):

Раздел №5

42. Как осуществляются сужающие и расширяющие приведения на целочисленной группе типов?
43. Как осуществляются сужающие и расширяющие приведения на вещественной группе типов?
44. Произойдет ли потеря точности при следующем преобразовании:
    float f = -16777217;
    Ответ пояснить.
45. Проанализировать код на предмет синтаксической корректности. Если код корректен, то какие значения примут все переменные, если нет, то какие ошибки будут сгенерированы:
    1) byte a = 100 - 100;

2) byte b = 100 + 100;
3) byte c = 100 * 100;

46. Верны ли следующие выражения для переменной d вещественного типа (float или double):
    1) (short)d==(short)(int)d;
    2) (int)d==(int)(long)d;
47. Корректны ли следующие преобразования:
    1) Object o = (String)null;
    2) String s = o;
48. Пусть классы Wolf и Rabbit являются наследниками класса Animal. Корректен ли следующий программный код:

```
Wolf  w = new Wolf();
Animal  a = (Animal)w;
Rabbit  r = (Rabbit)a;
```

Раздел №6

49. Перечислите основные интерфейсы, используемые при работе с коллекциями?
50. Реализация какого интерфейса позволяет сравнивать экземпляры класса друг с другом и сортировать их, например, в коллекциях?
51. Выберите верные утверждения:
    1) в коллекциях содержатся ссылки на объекты, а не их копии
    2) классы-коллекции, реализующие интерфейс List, поддерживают порядок элементов
    3) классы-коллекции, реализующие интерфейс Set, не допускают дублирования элементов
    4) классы-коллекции, реализующие интерфейс Map, используют уникальные ключи для размещения и поиска элементов
    5) в коллекциях могут сохранятся ссылки как на объекты, так и на примитивные типы
    6) ссылки на объекты в коллекциях Vector хранятся в порядке их добавления.
    7) в качестве ключей для коллекций типа Hashtable должны передаваться только объекты типа String
    8) нет верных утверждений
52. Какой из перечисленный ниже интерфейсов реализует класс HashTable:
    1) SortedMap
    2) Map
    3) List
    4) SortedSet
    5) нет верных утверждений

Раздел №7

53. Причины возникновения синхронных и асинхронных исключений?
54. Приведите пример программного кода порождения и обработки исключительных ситуаций с помощью try-catch-finally конструкции, операторов throw и throws. Пример пояснить
55. Что будет напечатано в результате выполнения программного кода:

```
public class Test {
        float fVal = 0.0f;
        public Test() {
        }
        public static void main(String[] args) {
                Test t = new Test();
                String testVal = "0.123";
```

```
                System.out.println("Was returned " + t.testParse(testVal) + "
                        with value " + t.fVal);
                }
                private boolean testParse(String val){
                try {
                        fVal = Float.parseFloat(val);
                        return true;
                }catch (NumberFormatException ex) {
                        System.out.println("Test.testParse() Bad number -> " + val);
                        fVal = Float.NaN;
                } finally{
                        System.out.println("Finally part executed");
                }
                return false;
        }
}
```

Раздел №8

56. Какие источники могут быть использованы классами стандартных входных потоков Java в качестве источника данных?
57. От какого класса наследуются InputStream и OutputStream? Остальные классы потоков ввода/вывода?
58. Что произойдет, если, используя объект типа ObjectOutputStream, записать в файл данные в порядке long, int, byte, а считать, используя объект типа DataInputStream, - в порядке byte, int, long?
59. Какая кодировка используется классом OutputStreamWriter по умолчанию?
60. Какие методы объявлены в интерфейсе Serializable?

Раздел №9

61. Какие из перечисленных ниже характеристик относятся к протоколу TCP:
    1) образование постоянного соединения
    2) не гарантирует доставку сообщения
    3) протокол уровня приложения
    4) использование в важных сервисах
62. Какие утилиты используются для тестирования работоспособности сети в ОС Windows?
63. Какие классы используются для работы с TCP-протоколом?
64. Можно ли с помощью класса URL пересылать данные на сервер?

Раздел №10

65. Для чего служит в Java класс Thread:
    1) для запуска потоков;
    2) для остановки потоков;
    3) для синхронизации потоков;
    4) для изменения свойств потоков (н-р, приоритетов)
66. Каким образом на однопроцессорной машине исполняются многопоточные приложения?
67. Какие преимущества дает многопоточная архитектура?
68. Какое преимущество дает использование интерфейса Runnable перед прямым наследованием класса Thread?

Раздел №11

69. Как получить объект класса Color, описывающий чистый синий цвет:

1) new Color (“blue”);
2) new Color (0,0,255);
3) Color.getBlue();
4) Color.blue

70. В чем разница между компонентами List и Choice?
71. Какой метод нужно переопределить, чтобы реализовать отрисовку внешнео вида компонента?
72. Какими параметрами в Java характеризуется шрифт?
73. Перечислите основные события AWT
74. Какой интерфейс и какой метод нужно реализовать, чтобы обработать событие нажатия кнопки?

Рейтинг-контроль №3:

На третий рейтинг-контроль выносятся материалы лабораторных работ, выполняемых студентами в течении семестра. **Контрольным мероприятием рейтинг-конроля являются отчеты по лабораторным работам**. В зависимости от результатов выполнения лабораторных работ и качества предоставленных отчетов студенту выставляется балл третьего рейтинг-контроля.

Лабораторные работы:

1. Объектная модель Java
2. Исключения
3. Коллекции
4. Потоки данных(stream), пакет java.io
5. Работа с сетью, пакет java.net
6. Потоки выполнения
7. Пользовательский интерфейс, пакет java.awt

Самостоятельная работа:

В часы, отведенные на самостоятельную работу по курсу, студенты выполняют индивидуальное семестровое задание. **Контрольным мероприятием для оценки выполнения студентом самостоятельной работы является итоговый отчет по семестровой работе**

Студентам предлагается разработать полноценное приложение в соответствии с индивидуальным заданием.

Требования к выполнению

1. Проектирование и реализация объектной модели
2. Реализация серверной и клиентской части: данные размещаются на сервере, вся бизнес-логика выполняется так же на сервере, клиентское приложение через пользовательский интерфейс отображает данные объектной модели, принимает запросы пользователя и отправляет эти запросы на сервер для обработки
3. В качестве хранилища данных - файловая система (все данные сохраняются в файловой системе)
4. Задача должна быть реализована с соблюдением основных принципов ООП (инкапсуляция, наследование, полиморфизм) и строго в концепции Model-View-Controller (отделение на уровне программного кода представления от бизнес-логики)

Задачи могут выполняться как индивидуально, так и в небольших группах по 2-3 человека.

Список задач

1. File Server. Сервер предоставляет свою файловую систему. Клиент может просматривать файловую систему сервера и обмениваться с сервером файлами (скачивать и закачивать файлы с сервера \ на сервер). Задача не предполагает регистрацию клиента. Раздел файловой системы сервера, открытый для клиента, определятся ip-адресом клиента (например, если ip-адрес клиента лежит в диапазоне 100.100.50.*, то клиент доступен раздел ФС сервера: - D:\documents).

1-2 человека

2. File Server. Сервер предоставляет клиентам информационные ресурсы для скачивания (фильмы, фотографии, музыку и проч). Для получения доступа к ресурсам клиенту необходимо завести на сервере свой аккаунт и получить статус – простой пользователь, вип-пользователь. От статуса будет зависеть скорость скачивания. Каждая сессия клиента начинается с процедуры авторизации.
   2-3 человека
3. Клиент-серверный чат
   1-2 человека
4. Сервер предоставляет клиенту HTML-страницы. Клиент скачивает с сервера эти страницы, находит по HTML-странице все ссылки, скачивает все связанные страницы и закачивает все найденные ресурсы на сервер.
   1-2 человека
5. Сервер рассылает сигналы точного времени, клиент синхронизируюет свое время.
   1 человек
6. Мониторинг рабочего времени персонала. Сервер осуществляет мониторинг рабочего времени сотрудников: фиксирует время прихода\ухода на рабочее место; прихода\ухода на обеденный перерыв; а так же периодически в основное рабочее время посылает контрольный вопрос сотруднику, для фиксации его фактического присутствия на рабочем месте. Результаты мониторинга для каждого сотрудника записываются в отдельный файл
   1-2 человека
7. Система учета студентов (картотека), сервер на основе файловой системы.
   1-2 человека
8. Сапер
   1-2 человека
9. Пятнашки
   1-2 человека
10. Тетрис
    1-2 человека
11. Морской бой: клиент с клиентом. Сервер выполняет функцию авторизации пользователей, предоставляет список возможных соперников, организует обмен данными между игроками в процессе игры, осуществляет мониторинг игры и сохранение состояния игры в случае ее досрочного завершения.
    1-2 человека
12. Морской бой: клиент с сервером. Сервер выполняет функцию авторизации пользователей, реализует логику соперника и сохранение состояния игры в случае ее досрочного завершения.
    2-3 человека
13. Крестики-нолики: клиент с клиентом. Сервер выполняет функцию авторизации пользователей, предоставляет список возможных соперников, организует обмен данными между игроками в процессе игры, осуществляет мониторинг игры и сохранение состояния игры в случае ее досрочного завершения.
    1-2 человек
14. Крестики-нолики: клиент с сервером. Сервер выполняет функцию авторизации пользователей, реализует логику соперника и сохранение состояния игры в случае ее досрочного завершения.
    2-3 человека
15. Томагочи \ Виртуальный Зоопарк \ Виртуальный заповедник
    1 человек

ПРОМЕЖУТОЧНАЯ АТТЕСТАЦИЯ ПО ИТОГАМ ОСВОЕНИЯ ДИСЦИПЛИНЫ
(зачет с оценкой)

Промежуточная аттестация проводится в форме зачета с оценкой. Итоговая оценка формируется как сумма баллов, набранных студентом в течение семестра и балла, полученного на экзамене. Максимальное количество баллов по всем видам работ в семестре указано в *Таблице 1*

Таблица 1

№	Наименование работ	Максимальное количество баллов по видам работ
1	Рейтинг-контроль 1	15
2	Рейтинг-контроль 2	15
3	Рейтинг-контроль 3	30
4	Выполнение семестрового плана самостоятельной работы – итоговый отчет по семестровой работе	30
5	Посещение занятий	5
6	Дополнительный баллы	5
Итого (максимум)		100

Вопросы к зачету с оценкой:

1. Платформа J2SE: основные пакеты
2. Лексемы, операторы, операции
1. Базовые примитивные типы данных
2. Базовые ссылочные типы данных
3. Приведение примитивных типов: расширяющие, сужающие, тождественные
4. Приведение ссылочных типов
5. Запрещенные приведения типов
6. Массивы примитивных и ссылочных типов
7. Классы и объекты, конструкторы, статические элементы класса, модификаторы видимости
8. Наследование, абстрактные классы, интерфейсы
9. Механизм позднего связывания и полиморфизм
10. Классы java.lang.Object и java.lang.Class: назначение, основные методы
11. Понятие исключительной ситуации, причины возникновения, классификация, обработка
12. Пользовательские классы исключений
13. Потоки данных, сериализация
14. Работа с сетью, основные классы пакета java.net
15. Потоки выполнения, касс java.lang.Thread, итерфейс java.lang.Runnable;
16. Многопоточная архитектура в клиент-серверных приложениях
17. Архитектурный шаблон проектирования MVC

7. УЧЕБНО-МЕТОДИЧЕСКОЕ И ИНФОРМАЦИОННОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

а) основная литература:

1. Васюткина И.А. Технология разработки объектно-ориентированных программ на JAVA [Электронный ресурс]: учебно-методическое пособие/ Васюткина И.А.— Электрон. текстовые данные.— Новосибирск: Новосибирский государственный технический университет, 2012.— 152 с.— Режим доступа: http://www.iprbookshop.ru/45047.— ЭБС «IPRbooks», по паролю

2. Задачи по программированию [Электронный ресурс]/ С.М. Окулов [и др.].— Электрон. текстовые данные.— М.: БИНОМ. Лаборатория знаний, 2014.— 824 с.— Режим доступа:http://www.studentlibrary.ru/book/ISBN9785996302529.html — ЭБС «Консультант студента», по паролю
3. Златопольский Д.М. Программирование. Типовые задачи, алгоритмы, методы [Электронный ресурс]/ Златопольский Д.М.— Электрон. текстовые данные.— М.: БИНОМ. Лаборатория знаний, 2015.— 224 с.— Режим доступа:http://www.studentlibrary.ru/book/ISBN9785996329328.html— ЭБС «Консультант студента», по паролю
4. Фарафонов А.С. Программирование на языке высокого уровня [Электронный ресурс]: методические указания к проведению лабораторных работ по курсу «Программирование»/ Фарафонов А.С.— Электрон. текстовые данные.— Липецк: Липецкий государственный технический университет, ЭБС АСВ, 2013.— 32 с.— Режим доступа: http://www.iprbookshop.ru/22912.— ЭБС «IPRbooks», по паролю

б) дополнительная литература:

1. Лисицин Д.В. Объектно-ориентированное программирование [Электронный ресурс]: конспект лекций/ Лисицин Д.В.— Электрон. текстовые данные.— Новосибирск: Новосибирский государственный технический университет, 2010.— 88 с.— Режим доступа: http://www.iprbookshop.ru/44970.— ЭБС «IPRbooks», по паролю
2. Бабушкина И.А. Практикум по объектно-ориентированному программированию [Электронный ресурс]/ Бабушкина И.А., Окулов С.М.— Электрон. текстовые данные.— М.: БИНОМ. Лаборатория знаний, 12— 367 с.— Режим доступа: http://www.studentlibrary.ru/book/ISBN9785996309542.html— ЭБС «Консультант студента», по паролю
3. Кауфман В.Ш. Языки программирования. Концепции и принципы [Электронный ресурс]/ Кауфман В.Ш.— Электрон. текстовые данные.— М.: ДМК Пресс, 2010.— 464 с.— Режим доступа: http://www.studentlibrary.ru/book/ISBN9785940746225.html.— ЭБС «Консультант студента», по паролю
4. Задачи по программированию [Электронный ресурс] / С.М. Окулов [и др.] ; под ред. С.М. Окулова. - 2-е изд., испр. (эл.). - М. : БИНОМ, 2014. .— Режим доступа: http://www.studentlibrary.ru/book/ISBN9785996323722.html.— ЭБС «Консультант студента», по паролю

в) интернет-ресурсы:

1. Вязовик Н.А. Программирование на Java. Режим доступа: http://www.intuit.ru/studies/ courses/16/16/info.
2. The Java® Language Specification, Java SE 7 Edition.. James Gosling, Bill Joy, Guy Steele, Gilad Bracha, Alex Buckley. Режим доступа: http://docs.oracle.com/javase/specs/jls/se7/ html/index.html.
3. The Java® Virtual Machine Specification, Java SE 7 Edition. Tim Lindholm, Frank Yellin, Gilad Bracha, Alex Buckley. Режим доступа: http://docs.oracle.com/javase/specs/jvms/se7/ html/index.html.
4. Java™ Platform, Standard Edition 7, API Specification. Режим доступа: http:// docs.oracle.com/javase/7/docs/api/index.html
5. Java™ Tutorials. Режим доступа: http://docs.oracle.com/javase/tutorial/index.html

г) программное обеспечение (ПО):

1. Комплект разработки приложений Java 2 SDK, Standart Edition, v1.7 (JDK). Режим для скачивания и установки:

http://www.oracle.com/technetwork/java/javase/downloads/jdk7-downloads-1880260.html

2. Среда исполнения приложений Java 2 Runtime Environment, Standart Edition, v 1.7 (JRE). Режим для скачивания и установки: http://www.oracle.com/technetwork/java/javase/downloads/jre7-downloads-1880261.html
3. Среда разработки программного кода Eclipce v 4.2.2. Режим для скачивания и установки: http://www.eclipse.org/downloads/packages/eclipse-classic-422/junosr2

Все ПО из списка является бесплатным, находится в открытом доступе и загружается для последующей установки с соответствующих сайтов

2. МАТЕРИАЛЬНО-ТЕХНИЧЕСКОЕ ОБЕСПЕЧЕНИЕ ДИСЦИПЛИНЫ (МОДУЛЯ)

Лекционные аудитории, оснащённые доской (для мела или маркера), экраном для проекционных систем, проектором и ноутбуком (420-3, 430-3).

Аудитории для проведения лабораторных занятий, оснащённые современными персональными компьютерами, объединёнными в локальную вычислительную сеть и укомплектованными необходимым системным и прикладным программным обеспечением (122б-3, 100-3, 511-3), аудитории вычислительного центра.

Рабочая программа дисциплины составлена в соответствии с требованиями ФГОС ВО по направлению 02.03.03 «Математическое обеспечение и администрирование информационных систем».

Рабочую программу составил ст. препод. кафедры ФиПМ Воронова Н.М. _____

Рецензент

(представитель работодателя) Квасов Д.С. Гл. инженер и ООО «Ре-Сервис»

(место работы, должность, ФИО, подпись)

Программа рассмотрена и одобрена на заседании кафедры ФиПМ

Протокол № 11а от 15.04.15 года

Заведующий кафедрой _____

(ФИО, подпись)

Рабочая программа рассмотрена и одобрена на заседании учебно-методической комиссии направления 02.03.03 «Математическое обеспечение и администрирование информационных систем»

Протокол № 11а от 15.04.15 года

Председатель комиссии _____

(ФИО, подпись)

ЛИСТ ПЕРЕУТВЕРЖДЕНИЯ
РАБОЧЕЙ ПРОГРАММЫ ДИСЦИПЛИНЫ (МОДУЛЯ)

Рабочая программа одобрена на _____ учебный год

Протокол заседания кафедры № _____ от _____ года

Заведующий кафедрой _____

Рабочая программа одобрена на _____ учебный год

Протокол заседания кафедры № _____ от _____ года

Заведующий кафедрой _____

Рабочая программа одобрена на _____ учебный год

Протокол заседания кафедры № _____ от _____ года

Заведующий кафедрой _____