



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Магнитогорский государственный технический университет им. Г.И. Носова»



УТВЕРЖДАЮ:  
Директор института  
горного дела и транспорта  
С.Е. Гавришев  
«07» сентября 2018 г.

## РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

### Б1.В.13 ОСНОВЫ ИНФОРМАТИКИ В ТРАНСПОРТНЫХ ПРИЛОЖЕНИЯХ

Направление подготовки  
38.03.02 Менеджмент

Профиль программы  
Логистика

Уровень высшего образования – бакалавриат  
Программа подготовки – прикладной бакалавриат


Форма обучения  
заочная

Институт	Горного дела и транспорта
Кафедра	Логистики и управления транспортными системами
Курс	3

Магнитогорск  
2018 г.

Рабочая программа составлена на основе ФГОС ВО по направлению подготовки 38.03.02 Менеджмент, утвержденного приказом МОиН РФ от 12.01.2016 № 7.

Рабочая программа рассмотрена и одобрена на заседании кафедры логистики и управления транспортными системами «06» сентября 2018г., протокол № 1.


Зав. кафедрой  / С.Н. Корнилов /  
(подпись) (И.О. Фамилия)

Рабочая программа одобрена методической комиссией института горного дела и транспорта «07» сентября 2018 г., протокол № 1.

Председатель  / С.Е. Гавришев /  
(подпись) (И.О. Фамилия)

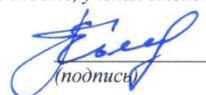
Рабочая программа составлена:

доцент, к.т.н. кафедры ЛиУТС  
(должность, ученая степень, ученое звание)

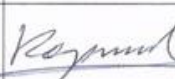
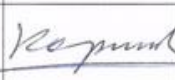
 / П.Н. Мишуров /  
(подпись) (И.О. Фамилия)

Рецензент:

ведущий инженер-технолог ПТГ УЛ ПАО «ММК»  
(должность, ученая степень, ученое звание)

 / Е.В. Полежаев /  
(подпись) (И.О. Фамилия)

### Лист регистрации изменений и дополнений

№ п/п	Раздел программы	Краткое содержание изменения/дополнения	Дата. № протокола заседания кафедры	Подпись зав. кафедрой
1	8	Актуализация учебно-методического и информационного обеспечения дисциплины (модуля)	03.09.2019г., протокол №1	
2	8	Актуализация учебно-методического и информационного обеспечения дисциплины (модуля)	01.09.2020г., протокол №1	

## 1 Цели освоения дисциплины (модуля)

Целями освоения дисциплины (модуля) является развитие у студентов личностных качеств, а также формирование общекультурных и профессиональных компетенций в области информатики для решения теоретических и практических задач по вопросам повышения эффективности функционирования производственных и транспортных систем на основе использования транспортных приложений.

## 2 Место дисциплины (модуля) в структуре образовательной программы подготовки бакалавра (магистра, специалиста)

Дисциплина входит в вариативную часть блока 1 образовательной программы.

Для изучения дисциплины необходимы знания (умения, навыки), сформированные в результате изучения следующих дисциплин:

- Информатика.
- Управление транспортными системами.
- Основы логистики и управление запасами в цепях поставок.

Знания (умения, навыки), полученные при изучении данной дисциплины будут необходимы при изучении следующих специальных дисциплин:

- Основы баз данных.
- Системный анализ в логистике.
- Информационные системы в логистике.
- Имитационное моделирование транспортных систем.
- Агентное моделирование транспортных систем.
- Разработка веб-сайтов.
- Современные интернет-технологии.

## 3 Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля) и планируемые результаты обучения

В результате освоения дисциплины (модуля) «Основы информатики в транспортных приложениях» обучающийся должен обладать следующими компетенциями:

Структурный элемент компетенции	Планируемые результаты обучения
<b>ПК-10 владение навыками количественного и качественного анализа информации при принятии управленческих решений, построения экономических, финансовых и организационно-управленческих моделей путем их адаптации к конкретным задачам управления</b>	
Знать	– виды и типы языков программирования, их характеристики; – технологические характеристики основных языков программирования; – особенности использования различных языков программирования в современных системах управления на транспорте.
Уметь	– структурировать значительные объемы статистической информации об объекте реального мира; – создавать программы на основании обработки информационных процессов; – создавать и использовать транспортные приложения
Владеть	– способами системами сбора, обработки и хранения информации; – методами обобщения, разделения и анализа данных о транспортном объекте; – методиками программирования процесса управления перевозочным процессом.

Структурный элемент компетенции	Планируемые результаты обучения
<b>ПК-11 владение навыками анализа информации о функционировании системы внутреннего документооборота организации, ведения баз данных по различным показателям и формирования информационного обеспечения участников организационных проектов</b>	
Знать	<ul style="list-style-type: none"> <li>– показатели и характеристики информационных потоков в транспортных системах;</li> <li>– основные инструменты написания программ;</li> <li>– основы языка программирования -java;</li> </ul>
Уметь	<ul style="list-style-type: none"> <li>– выделять требуемые характеристики объекта для написания программного кода;</li> <li>– структурировать программный код при формировании транспортного приложения;</li> <li>– адаптировать разработанные программы к условия функционирования транспортного предприятия;</li> </ul>
Владеть	<ul style="list-style-type: none"> <li>– методами обработки данных о программируемом объекте;</li> <li>– методами описания информационных процессов на алгоритмическом языке;</li> <li>– программными инструментами цифровизации объектов транспорта .</li> </ul>

#### 4 Структура и содержание дисциплины (модуля)

Общая трудоемкость дисциплины составляет 3 зачетных единиц 108 академических часов, в том числе:

- контактная работа – 8,4 академических часов:
  - аудиторная – 8, академических часов;
  - внеаудиторная – 0,4 академических часов
- самостоятельная работа – 95,7 академических часов;
- подготовка к зачету – 3,9 академических часов.

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в академических часах)			Самостоятельная работа (в академических часах)	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код и структурный элемент компетенции
		лекции	лаборат. занятия	практич. занятия				
1 Раздел «Программное обеспечение и вычислительной техники и его классификация»	3							
1.1 Тема «Основные понятия теории информации»		0,1		0,3	5,3	Поиск дополнительной информации по заданной теме	Устный опрос	ПК-10 - зув
1.2 Тема «Тенденции развития программного обеспечения для ЭВМ и систем коммуникации»		0,2		0,5	5,3	Поиск дополнительной информации по заданной теме	Устный опрос	ПК-10 - зув ПК-11 - зув
1.3 Тема «Классификация базового и прикладного программного обеспечения»		0,2		0,5	5,3	Поиск дополнительной информации по заданной теме	Устный опрос	ПК-10 - зув ПК-11 - зув
1.4 Тема «Примеры использования программного обеспечения в различных отраслях народного хозяйства»		0,1		0,3	5,3	Самостоятельное изучение учебной и научной литературы	Устный опрос	ПК-10 - зув ПК-11 - зув
Итого по разделу	3	0,6		1,6	21,2		Устный опрос	
2 Раздел «Объектно-ориентированный	3							

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа (в акад. часах)	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код и структурный элемент компетенции
		лекции	лаборат. занятия	практич. занятия				
подход к компьютерному программированию»								
2.1 Тема «Основы объектно-ориентированный подхода к компьютерному программированию»		0,1		0,3	5,3	Самостоятельное изучение учебной и научной литературы	Устный опрос	ПК-10 - зув ПК-11 - зув
2.2 Тема «Типы данных Java и оператор if»		0,1		0,3	5,3	Подготовка к практическим занятиям, выполнение практических работ	Проверка практических заданий, устный опрос.	ПК-10 - зув
2.3 Тема «Массивы»		0,1		0,3	5,3	Подготовка к практическим занятиям, выполнение практических работ	Проверка практических заданий, устный опрос	ПК-11 - зув
2.4 Тема «Циклы»		0,1		0,3	5,3	Подготовка к практическим занятиям, выполнение практических работ	Проверка практических заданий, устный опрос.	ПК-11 - зув
2.5 Тема «Коллекции»		0,1		0,3	5,3	Подготовка к практическим занятиям, выполнение практических работ	Проверка практических заданий, устный опрос.	ПК-10 - зув ПК-11 - зув
2.6 Тема «Операторы сравнения, логические операторы и switch»		0,1		0,3	5,3	Подготовка к практическим занятиям, выполнение практических работ	Проверка практических заданий, устный опрос.	ПК-10 - зув ПК-11 - зув
2.7 Тема «Видимость и связи»		0,1		0,3	5,3	Подготовка к практическим занятиям, выполнение практических работ	Проверка практических заданий, устный опрос.	ПК-10 – зув ПК-11 - зув

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа (в акад. часах)	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код и структурный элемент компетенции
		лекции	лаборат. занятия	практич. занятия				
2.8 Тема «Взаимодействие объектов в AnyLogic»		0,1		0,2	5,3	Подготовка к практическим занятиям, выполнение практических работ	Проверка практических заданий, устный опрос.	ПК-10 – зув ПК-11 - зув
2.9 Тема «Диаграммы состояний»		0,1		0,1	5,3	Подготовка к практическим занятиям, выполнение практических работ	Проверка практических заданий, устный опрос.	ПК-10 – зув ПК-11 - зув
Итого по разделу	3	0,9		2,4	47,7		Устный опрос	
3 Раздел «Имитационное моделирование транспортных систем»	3			0,3	5,3			
3.1 Тема «Основные понятие метода имитационного моделирования»		0,1		0,3	5,3	Поиск дополнительной информации по заданной теме	Устный опрос	ПК-11 - зув
3.2 Тема «Системно-динамический подход к построению имитационных моделей»		0,1		0,3	5,3	Поиск дополнительной информации по заданной теме	Устный опрос	ПК-11 - зув
3.3 Тема «Дискретно-событийный и агентный подходы к построению имитационных моделей»		0,1		0,3	5,4	Поиск дополнительной информации по заданной теме	Устный опрос	ПК-11 - зув
3.4 Тема «Особенности построения имитационных моделей транспортных систем»		0,1		0,4	5,4	Самостоятельное изучение учебной и научной литературы	Устный опрос	ПК-10 – зув ПК-11 - зув
3.5 Тема «Универсальные системы построения имитационных моделей. Инструмент имитационного моделирования AnyLogic»		0,1		0,4	5,4	Поиск дополнительной информации по заданной теме	Устный опрос	ПК-10 – зув ПК-11 - зув
Итого по разделу	3	0,5		2	26,8		Устный опрос	



Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа (в акад. часах)	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код и структурный элемент компетенции
		лекции	лаборат. занятия	практич. занятия				
<b>Итого по дисциплине</b>	<b>3</b>	<b>2</b>		<b>6</b>	<b>95,7</b>		<b>Зачёт</b>	

## 5 Образовательные и информационные технологии

Для реализации предусмотренных видов учебной работы в качестве образовательных технологий в преподавании дисциплины «Основы информатики в транспортных приложениях» используются традиционные интерактивная и модульно-компетентностная технологии.

В ходе проведения лекционных и практических занятий предусматривается:

- использование электронного демонстрационного материала по темам, требующим иллюстрации работы специализированного программного обеспечения, сложных структурных схем и большого объема графического материала;

- активные и интерактивные формы обучения: вариативный опрос, дискуссии, устный опрос, разбор конкретных ситуаций и т.д.

Образовательные технологии в сочетании с внеаудиторной работой нацелены на формирование и развитие профессиональных навыков обучающихся.

Самостоятельная работа стимулирует студентов в процессе подготовки домашних заданий, при подготовке к итоговой аттестации, которая осуществляется в форме защиты подготовленных рефератов.

## 6 Учебно-методическое обеспечение самостоятельной работы обучающихся

По дисциплине «Основы информатики в транспортных приложениях» предусмотрена аудиторная и внеаудиторная самостоятельная работа обучающихся.

Аудиторная самостоятельная работа студентов на практических занятиях осуществляется под контролем преподавателя в виде решения задач и выполнения упражнений, которые определяет преподаватель для студента.

Внеаудиторная самостоятельная работа обучающихся осуществляется в виде изучения литературы по соответствующему разделу с проработкой материала; выполнение практических заданий.

Перечень практических заданий:

### Практическое задание №1 на тему «Типы данных»

Общие правила приведения типов переменных Java в выражениях:

- типы переменных `byte`, `short` и `char` повышаются до типа `int`;
- разрядность целевой переменной (выражения) соответствует максимальной разрядности операнда (переменной выражения).

Пример корректного использования типа целевой переменной:

```
byte b = 45;
char c = 'c';
short s = 1005;
int i = 700000;
float f = 4.55f;
double d = 1.456;
double result = (f * b) + (i / c) - (d * s);
System.out.println("result равен " + result);
```

### Практическое задание №2 на тему «Логические операторы»

Пример вложенного оператора `if`:

```
if(i == 10)
{
    if(j < 20) a = b;
    if(k > 100) c = d;
    else a = c; // else относится к if(k > 100)
}
else a = d; // else относится к if(i == 10)
```

Конструкция if-else

```
intlargerNum;  
intlowNum = 9;  
inthighNum = 27;  
  
if(lowNum<highNum) // если первое число меньше второго  
{  
largerNum = highNum;  
} else{ // иначе  
largerNum = lowNum;  
}
```

Тернарный оператор

```
int lowNum = 9;  
int highNum = 27;  
int largerNum = lowNum<highNum ? highNum :lowNum;
```

```
int absval, val;  
val = 5;  
absval = val< 0 ? -val:val;  
// выводим число  
System.out.println("" + absval);  
val = -5;  
absval = val< 0 ? -val:val;  
System.out.println("" + absval);
```

Пример использования оператора switch– программа для получения кода типа вагона.

```
Stringвагон = "полувагон";  
intкод = 0;  
switch (вагон) {  
case"крытый":код = 2;  
break;  
case"платформа":код = 4;  
break;  
case"полувагон":код = 6;  
break;  
case"цистерна":код = 7;  
break;  
case"изотермический":код = 8;  
break;  
case"прочие":код = 9;  
break;  
default: System.out.println("Неверныйтипвагона");  
break;  
}  
if (код != 0) System.out.println("Типвагона: " + вагон + ",код: " + код);
```

### Практическое задание №3 на тему «Массивы»

Пример одновременного создания и инициализации массива в AnyLogic (запись помещается в поле «Начальное значение» переменной)

```
newint[] { 13, x-3, -15, 0, max(a,100) };  
или в программе на языкеJava  
int[] intarray = new int[] { 13, x-3, -15, 0, max( a, 100 ) };
```

Пример создания массива строковых переменных  
String[] список = new String[] { “Света”, “Таня”, “Маша” };

Двумерный массив значений типа double  
double[][] двумерныйМассив = new double[10][20];  
int числоСтрок = двумерныйМассив.length;  
int числоСтолбцов = двумерныйМассив[0].length;  
System.out.println(“Двумерный массив содержит ”+ числоСтрок+ ” строк и ” + числоСтолбцов + ” столбцов”);

При инициализации многомерных массивов используют дополнительные фигурные скобки, например, в программе на языке Java

```
int[][] intarrayD = {  
    { 1, 2, 3 },  
    { 4, 5, 6 }  
}
```

или при указании начального значения переменной-массива в AnyLogic

```
newint[][] {{ 1, 2, 3 }, { 4, 5, 6 }}
```

Данный массив будет содержать две строки, в каждой из которых находится три элемента.

Представление созданного в предыдущем примере массива форме таблицы

Индексы строк	Индексы столбцов		
	0	1	2
0	1	2	3
1	4	5	6

Элемент массива intarrayD[1][2], например, будет равен 6.

#### Практическое задание №4 на тему «Циклы»

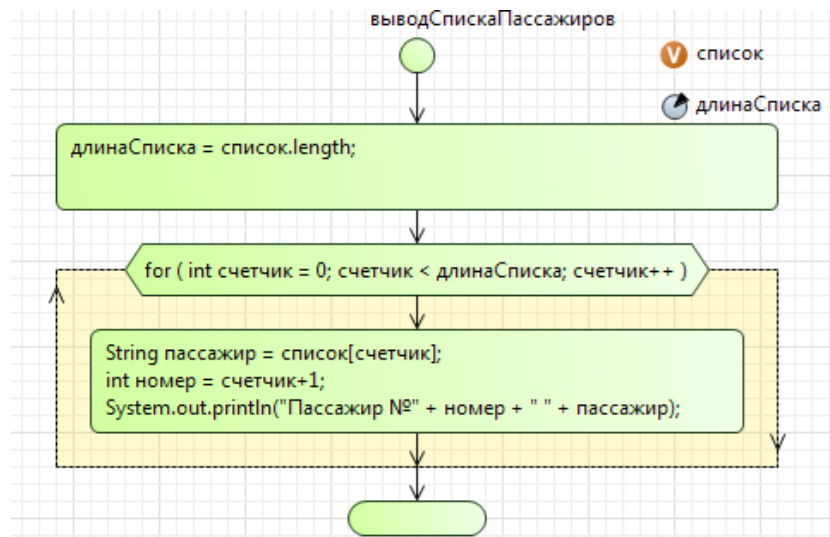
Пример цикла **for** на языке Java:

```
String[] список = new String[] { “Света”, “Таня”, “Маша”, “Дима” };  
int длинаСписка = список.length;  
int счетчик;  
for (счетчик = 0; счетчик < длинаСписка; счетчик++) {  
    String пассажир = список[счетчик];  
    int номер = счетчик + 1;  
    System.out.println(“Пассажир №” + номер + “” + пассажир);  
}
```

В результате выполнения цикла будут напечатано:

Пассажир №1 Света  
Пассажир №2 Таня  
Пассажир №3 Маша  
Пассажир №4 Дима

На следующем рисунке представлен пример создания цикла **for** при помощи «диаграммы действий» AnyLogic



## Цикл **while**

Проверка условия окончания цикла в цикле **while** происходит ДО начала выполнения цикла. Например, такой цикл выполнится четыре раза, а на экран будет выведено «1 2 3 4 »

```

inti = 1;
while (i < 5) {
    System.out.println(i + " ");
    i++;
}
  
```

При определённых условиях цикл **while** может ни разу не выполниться, например, поскольку значение *i* больше нуля, то тело данного цикла не выполнится и ничего напечатано не будет

```

inti = 1;
while (i < 0) {
    System.out.println(i + " ");
    i++;
}
  
```

Цикл **while** может выполняться бесконечное число раз, например

```

int i = 1;
while (true) {
    System.out.println(i + " ");
    i++;
}
  
```

## Цикл **do... while**

Условие окончания цикла в цикле **do ... while** происходит ПОСЛЕ выполнения тела цикла, поэтому в цикле **do ... while** операторы тела цикла выполняются как минимум один раз.

Например, такой цикл выполнится один раз, а на экран будет выведено «2 »

```

int i = 1;
do {
    i++;
    System.out.println(i + " ");
} while (i < 0);
  
```

Следующий цикл выполнится четыре раза, а на экран будет выведено «2 3 4 5 »

```
int i = 1;
do {
    i++;
    System.out.println(i + " ");
} while (i < 5);
```

### **Практическое задание №5** на тему «Коллекции»

Создать коллекцию `ArrayList`, инициализировать ее тридцатью элементами. Идентифицировать десятый и двадцатый элемент коллекции. Добавить в коллекцию 5 элементов. Удалить первый элемент коллекции и заменить одиннадцатый.

Методы коллекции `ArrayList`:

- `add(элемент)`– добавление элемента в конец коллекции. Элемент может переменной любого типа `Java`, экземпляром класса `Java` или агентом `AnyLogic`.
- `get(индекс элемента)` –получение элемента коллекции по его индексу. Нумерация индексов в коллекции начинается так же как и у массивов – с нуля.
- `size()` –получение числа элементов коллекции.
- `indexOf(элемент)` – получение индекса элемента в коллекции.
- `contains(элемент)`– определение наличия элемента коллекции. Если элемент присутствует, то результатом выполнения метода будут `true`, в противном случае – `false`.
- `remove(индекс)` или `remove(элемент)` – удаление элемента из коллекции, соответственно, по его индексу или по содержимому элемента.
- `set(индекс, элемент)` – замена элемента в коллекции с индексом «индекс» на новый элемент, указанный в качестве параметра метода `set`.
- `clear()` – очистка коллекции.

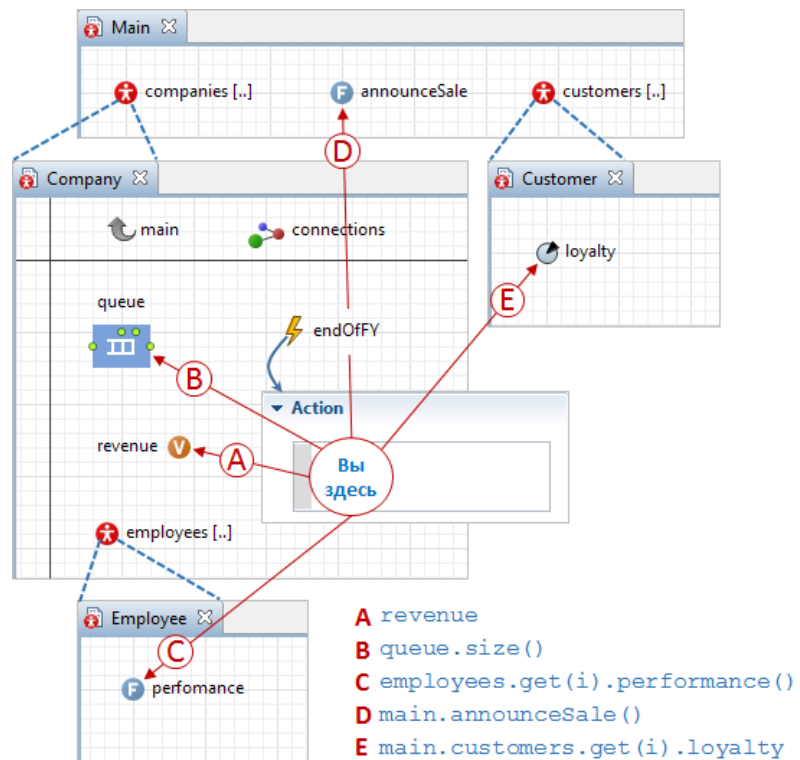
Создать коллекцию `LinkedList`, инициализировать ее тридцатью элементами. Идентифицировать десятый и двадцатый элемент коллекции. Добавить в коллекцию 5 элементов. Удалить первый элемент коллекции.

Дополнительные методы коллекции `LinkedList`:

- `addLast(элемент)` и `addFirst(элемент)` – добавление элемента, соответственно, в конец и начало списка.
- `add(индекс, элемент)` – добавление элемента в список ПЕРЕД элементом, с указанным индексом.
- `removeLast()` и `removeFirst()`–удаление элемента, находящегося, соответственно, в конце и начале списка.

### **Практическое задание №6** на тему «Взаимодействие объектов»

Получить доступ к другим элементам модели.



Объявить модификаторы.

```
public int i;
private double j, k;
private int createMethod(int a) {...};
public class Cat{}
```

## 7 Оценочные средства для проведения промежуточной аттестации

### а) Планируемые результаты обучения и оценочные средства для проведения промежуточной аттестации:

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
<b>ПК-10 владение навыками количественного и качественного анализа информации при принятии управленческих решений, построения экономических, финансовых и организационно-управленческих моделей путем их адаптации к конкретным задачам управления</b>		
Знать	<ul style="list-style-type: none"> <li>– виды и типы языков программирования, их характеристики;</li> <li>– технологические характеристики основных языков программирования;</li> <li>– особенности использования различных языков программирования в современных системах управления на транспорте.</li> </ul>	<p><b>Перечень теоретических вопросов:</b></p> <ol style="list-style-type: none"> <li>1. Понятие об имитационном моделировании. Назначение имитационных моделей.</li> <li>2. Современные подходы к имитационному моделированию. Сущность системно-динамического, дискретно-событийного (процессного) и агентного подходов к имитационному моделированию.</li> <li>3. Область применения системно-динамических имитационных моделей.</li> <li>4. Область применения дискретно-событийных (процессных) имитационных моделей.</li> <li>5. Область применения агентных имитационных моделей.</li> <li>6. Программные системы построения имитационных моделей. Их достоинства, недостатки и области применения. Достоинства мультиподходной системы AnyLogic.</li> <li>7. Особенности системы построения имитационных моделей AnyLogic. Рабочая область системы. Состав рабочей области программы. Палитры блоков.</li> <li>8. Сущность объектно-ориентированного подхода к разработке компьютерных программ.</li> <li>9. Понятие «абстракция» в объектно-ориентированном программировании. Примеры.</li> <li>10. Понятие «инкапсуляция» в объектно-ориентированном программировании. Примеры.</li> <li>11. Понятие «наследование» в объектно-ориентированном программировании. Примеры.</li> <li>12. Понятие «полиморфизм» в объектно-ориентированном программировании. Примеры.</li> </ol>
Уметь	<ul style="list-style-type: none"> <li>– структурировать значительные объемы статистической информации об объекте реального мира;</li> <li>– создавать программы на основании обработки информационных процессов;</li> </ul>	<p><b>Примерные практические задания:</b></p> <p style="text-align: center;"><b>Практическое задание №1</b> на тему «Типы данных»</p> <p>Общие правила приведения типов переменных Java в выражениях:</p> <ul style="list-style-type: none"> <li>• типы переменных byte, short и char повышаются до типа int;</li> <li>• разрядность целевой переменной (выражения) соответствует максимальной разрядности операнда (переменной выражения).</li> </ul> <p>Пример корректного использования типа целевой переменной:</p>



Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
	<p>– создавать и использовать транспортные приложения</p>	<pre>byte b = 45; char c = 'c'; short s = 1005; inti = 700000; float f = 4.55f; double d = 1.456; double result = (f * b) + (i / c) - (d * s); System.out.println("resultравен " + result);</pre> <p style="text-align: center;"><b>Практическое задание №2</b> на тему «Логические операторы»</p> <p>Пример вложенного оператора if:</p> <pre>if(i == 10) { if(j &lt; 20) a =b; if(k &gt; 100) c =d; else a = c; // else относится к if(k &gt; 100) } else a = d; // else относится к if(i == 10)</pre> <p>Конструкция if-else</p> <pre>intlargerNum; intlowNum = 9; inhighNum = 27;</pre> <pre>if(lowNum&lt;highNum) // если первое число меньше второго { largerNum = highNum; } else { // иначе largerNum = lowNum; }</pre>

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		<p>Тернарный оператор</p> <pre>int lowNum = 9; int highNum = 27; int largerNum = lowNum &lt; highNum ? highNum : lowNum;</pre> <p>int absval, val; val = 5; absval = val &lt; 0 ? -val : val; // выводим число System.out.println("" + absval); val = -5; absval = val &lt; 0 ? -val : val; System.out.println("" + absval);</p> <p>Пример использования оператора switch– программа для получения кода типа вагона.</p> <pre>String вагон = "полувагон"; int код = 0; switch (вагон) { case "крытый": код = 2; break; case "платформа": код = 4; break; case "полувагон": код = 6; break; case "цистерна": код = 7; break; case "изотермический": код = 8; break; case "прочие": код = 9;</pre>

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		<pre>break; default: System.out.println("Неверный тип вагона"); break; } if (код != 0) System.out.println("Тип вагона: " + вагон + ", код: " + код);</pre>
Владеть	<ul style="list-style-type: none"> <li>– способами системами сбора, обработки и хранения информации;</li> <li>– методами обобщения, разделения и анализа данных о транспортном объекте;</li> <li>– методиками программирования процесса управления перевозочным процессом.</li> </ul>	<p><b>Перечень теоретических вопросов:</b></p> <ol style="list-style-type: none"> <li>1. Системно-динамический подход к построению имитационных моделей</li> <li>2. Дискретно-событийный (процессный) подход к построению имитационных моделей</li> <li>3. Агентный подход к построению имитационных моделей.</li> </ol>
<b>ПК-11 владение навыками анализа информации о функционировании системы внутреннего документооборота организации, ведения баз данных по различным показателям и формирования информационного обеспечения участников организационных проектов</b>		
Знать	<ul style="list-style-type: none"> <li>– показатели и характеристики информационных потоков в транспортных системах;</li> <li>– основные инструменты написания программ;</li> <li>– основы языка программирования -java;</li> </ul>	<p><b>Перечень теоретических вопросов:</b></p> <ol style="list-style-type: none"> <li>1. Особенности языка программирования Java как языка объектно-ориентированного программирования. Направления использования программного кода на языке Java в среде AnyLogic.</li> <li>2. Типы данных в языке программирования Java.</li> <li>3. Оператор if в языке программирования Java. Примеры использования оператора if.</li> <li>4. Массивы в языке программирования Java. Примеры использования массивов.</li> <li>5. Циклы в языке программирования Java. Виды циклов. Примеры использования циклов.</li> <li>6. Коллекции в языке Java. Виды коллекций. Методы коллекций. Примеры использования коллекций.</li> <li>7. Операторы сравнения и логические операторы Java. Примеры использования операторов сравнения и логических операторов.</li> <li>8. Оператор switch в языке Java. Примеры использования оператора switch.</li> </ol>

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		<p>9. Правила видимости в языке Java и в системе AnyLogic. Примеры доступа к переменным и объектам в системе AnyLogic.</p> <p>10. Организация связей между агентами и взаимодействие агентов в системе AnyLogic. Примеры организации связей.</p> <p>11. Понятие «диаграммы состояний». Элементы диаграммы состояний. Порядок и примеры использования диаграмм состояний в системе AnyLogic.</p> <p>12. Диаграммы действий в системе AnyLogic. Назначение, порядок и примеры использования диаграмм действий.</p> <p>13. Системно-динамическое моделирование в AnyLogic. Особенности системно-динамических моделей. Примеры системно-динамических моделей.</p> <p>14. Основы планирования и проведения экспериментов с имитационными моделями в системе AnyLogic.</p>
Уметь	<ul style="list-style-type: none"> <li>– выделять требуемые характеристики объекта для написания программного кода;</li> <li>– структурировать программный код при формировании транспортного приложения;</li> <li>– адаптировать разработанные программы к условиям функционирования транспортного предприятия;</li> </ul>	<p><b>Примерные практические задания:</b></p> <p style="text-align: center;"><b>Практическое задание №3</b> на тему «Массивы»</p> <p>Пример одновременного создания и инициализации массива в AnyLogic (запись помещается в поле «Начальное значение» переменной)</p> <pre>newint[] { 13, x-3, -15, 0, max(a,100) }; или в программе на языкеJava int[] intarray = new int[] { 13, x-3, -15, 0, max( a, 100 ) }; Пример создания массива строковых переменных String[] список = newString[] { "Света", "Таня", "Маша" };</pre> <p>Двумерный массив значений типа double</p> <pre>double[][] двумерныйМассив = newdouble[10][20]; intчислоСтрок = двумерныйМассив.length; intчислоСтолбцов = двумерныйМассив[0].length; System.out.println("Двумерный массив содержит "+ числоСтрок+ " строк и " + числоСтолбцов + " столбцов");</pre>

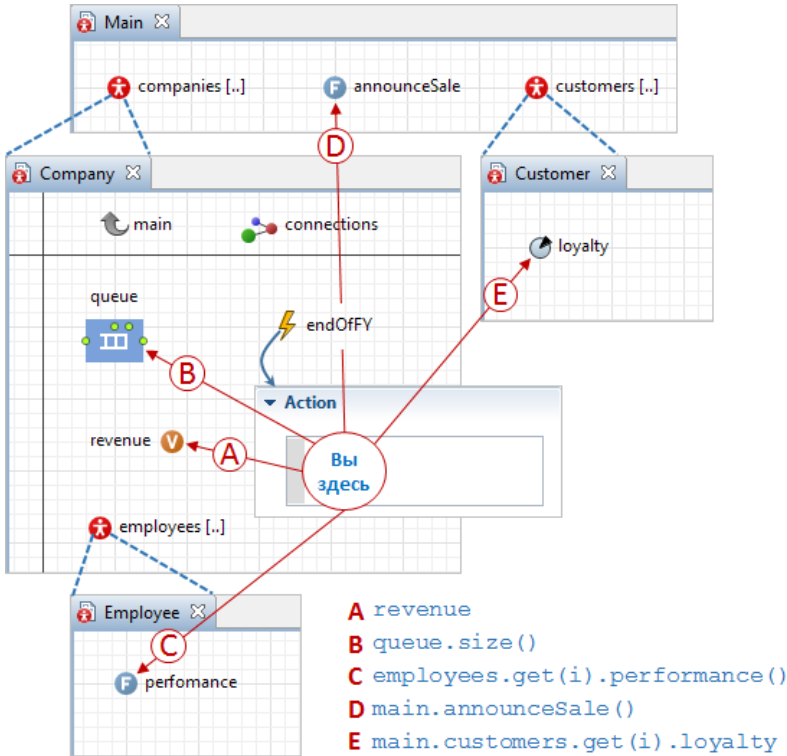
Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства															
		<p>При инициализации многомерных массивов используют дополнительные фигурные скобки, например, в программе на языке Java</p> <pre>int[][] intarrayD = {     { 1, 2, 3 },     { 4, 5, 6 } }</pre> <p>или при указании начального значения переменной-массива в AnyLogic</p> <pre>newint[][] {{ 1, 2, 3 },{ 4, 5, 6 }}</pre> <p>Данный массив будет содержать две строки, в каждой из которых находится три элемента. Представление созданного в предыдущем примере массива форме таблицы</p> <table border="1" data-bbox="1245 691 1789 847"> <thead> <tr> <th rowspan="2">Индексы строк</th> <th colspan="3">Индексы столбцов</th> </tr> <tr> <th>0</th> <th>1</th> <th>2</th> </tr> </thead> <tbody> <tr> <th>0</th> <td>1</td> <td>2</td> <td>3</td> </tr> <tr> <th>1</th> <td>4</td> <td>5</td> <td>6</td> </tr> </tbody> </table> <p>Элемент массива <code>intarrayD[1][2]</code>, например, будут равен 6.</p> <p style="text-align: center;"><b>Практическое задание №4</b> на тему «Циклы»</p> <p>Пример цикла <b>for</b> на языке Java:</p> <pre>String[] список = new String[] { "Света", "Таня", "Маша", "Дима" }; intдлинаСписка = список.length; intсчетчик; for (счетчик = 0; счетчик&lt;длинаСписка; счетчик++) {     String пассажир = список[счетчик];     int номер = счетчик+1;     System.out.println("Пассажир №" + номер + "" + пассажир); }</pre> <p>В результате выполнения цикла будут напечатано:</p> <pre>Пассажир №1 Света Пассажир №2 Таня</pre>	Индексы строк	Индексы столбцов			0	1	2	0	1	2	3	1	4	5	6
Индексы строк	Индексы столбцов																
	0	1	2														
0	1	2	3														
1	4	5	6														

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		<p>Пассажир №3 Маша Пассажир №4 Дима</p> <p>На следующем рисунке представлен пример создания цикла <b>for</b> при помощи «диаграммы действий» AnyLogic</p> <pre> graph TD     Start(( )) --&gt; Assign[длинаСписка = список.length;]     Assign --&gt; For{for ( int счетчик = 0; счетчик &lt; длинаСписка; счетчик++ )}     For --&gt; LoopBody[String пассажир = список[счетчик]; int номер = счетчик+1; System.out.println("Пассажир №" + номер + " " + пассажир);]     LoopBody --&gt; For     For --&gt; End([ ])   </pre> <p>Цикл <b>while</b></p> <p>Проверка условия окончания цикла в цикле while происходит ДО начала выполнения цикла. Например, такой цикл выполнится четыре раза, а на экран будет выведено «1 2 3 4 »</p> <pre> inti = 1; while (i &lt; 5) {     System.out.println(i + " ");     i++; }   </pre>

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		<p>При определённых условиях цикл <code>while</code> может ни разу не выполниться, например, поскольку значение <code>i</code> больше нуля, то тело данного цикла не выполнится и ничего напечатано не будет</p> <pre> inti = 1; while (i &lt; 0) {     System.out.println(i + " ");     i++; } </pre> <p>Цикл <code>while</code> может выполняться бесконечное число раз, например</p> <pre> int i = 1; while (true) {     System.out.println(i + " ");     i++; } </pre> <p><b>Цикл <code>do... while</code></b></p> <p>Условие окончания цикла в цикле <b><code>do ... while</code></b> происходит ПОСЛЕ выполнения тела цикла, поэтому в цикле <b><code>do ... while</code></b> операторы тела цикла выполняются как минимум один раз.</p> <p>Например, такой цикл выполнится один раз, а на экран будет выведено «2 »</p> <pre> int i = 1; do {     i++;     System.out.println(i + " "); } while (i &lt; 0); </pre> <p>Следующий цикл выполнится четыре раза, а на экран будет выведено «2 3 4 5 »</p> <pre> int i = 1; do {     i++;     System.out.println(i + " "); } while (i &lt; 5); </pre>

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		<pre data-bbox="875 352 1111 379">} while (i &lt; 5);</pre> <p data-bbox="1173 419 1861 446" style="text-align: center;"><b>Практическое задание №5</b> на тему «Коллекции»</p> <p data-bbox="875 456 2159 560">Создать коллекцию ArrayList, инициализировать ее тридцатью элементами. Идентифицировать десятый и двадцатый элемент коллекции. Добавить в коллекцию 5 элементов. Удалить первый элемент коллекции и заменить одиннадцатый.</p> <p data-bbox="875 568 1279 595">Методы коллекции ArrayList:</p> <ul data-bbox="875 608 2159 1098" style="list-style-type: none"> <li>– add(элемент)– добавление элемента в конец коллекции. Элемент может переменной любого типа Java, экземпляром класса Javaили агентом AnyLogic.</li> <li>– get(индекс элемента) –получение элемента коллекции по его индексу. Нумерация индексов в коллекции начинается так же как и у массивов – с нуля.</li> <li>– size() –получение числа элементов коллекции.</li> <li>– indexOf(элемент) – получение индекса элемента в коллекции.</li> <li>– contains(элемент)– определение наличия элемента коллекции. Если элемент присутствует, то результатом выполнения метода будут true, в противном случае – false.</li> <li>– remove(индекс)илиremove(элемент) – удаление элемента из коллекции, соответственно, по его индексу или по содержимому элемента.</li> <li>– set(индекс, элемент) – замена элемента в коллекции с индексом «индекс» на новый элемент, указанный в качестве параметра метода set.</li> <li>– clear() – очистка коллекции.</li> </ul> <p data-bbox="875 1142 2159 1246">Создать коллекцию LinkedList, инициализировать ее тридцатью элементами. Идентифицировать десятый и двадцатый элемент коллекции. Добавить в коллекцию 5 элементов. Удалить первый элемент коллекции.</p> <p data-bbox="875 1291 1525 1318">Дополнительные методы коллекции LinkedList:</p> <ul data-bbox="875 1331 2159 1436" style="list-style-type: none"> <li>– addLast(элемент) и addFirst(элемент) – добавление элемента, соответственно, в конец и начало списка.</li> <li>– add(индекс, элемент) – добавление элемента в список ПЕРЕД элементом, с указанным ин-</li> </ul>



Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		<p>дексом.</p> <p>– removeLast()и removeFirst()–удаление элемента, находящегося, соответственно, в конце и начале списка.</p> <p style="text-align: center;"><b>Практическое задание №6</b> на тему «Взаимодействие объектов»</p> <p>Получить доступ к другим элементам модели.</p>  <p><b>A</b> revenue  <b>B</b> queue.size()  <b>C</b> employees.get(i).performance()  <b>D</b> main.announceSale()  <b>E</b> main.customers.get(i).loyalty</p>

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		<p>Объявить модификаторы.</p> <pre>public int i; private double j, k; private int createMethod(int a) {...}; public class Cat {}</pre>
Владеть	<ul style="list-style-type: none"> <li>– методами обработки данных о программируемом объекте;</li> <li>– методами описания информационных процессов на алгоритмическом языке;</li> <li>– программными инструментами цифровизации объектов транспорта.</li> </ul>	<p><b>Перечень теоретических вопросов:</b></p> <ol style="list-style-type: none"> <li>1. Инструменты построения имитационных моделей.</li> <li>2. Методы работы с коллекциями Java.</li> <li>3. Методы работы с массивами данных.</li> </ol>

## **б) Порядок проведения промежуточной аттестации, показатели и критерии оценивания:**

Промежуточная аттестация по дисциплине «Основы информатики в транспортных приложениях» включает теоретические вопросы, позволяющие оценить уровень усвоения обучающимися знаний, и практические и комплексные задания, выявляющие степень сформированности умений и владений, проводится в форме зачёта.

Зачет по данной дисциплине проводится в устной форме.

### **Показатели и критерии оценивания зачета:**

– на оценку «зачтено» – обучающийся демонстрирует достаточный уровень сформированности компетенций, основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.

– на оценку «не зачтено» – обучающийся демонстрирует знания не более 20% теоретического материала, допускает существенные ошибки, не может показать интеллектуальные навыки решения простых задач.

## **8 Учебно-методическое и информационное обеспечение дисциплины (модуля)**

### **а) Основная литература:**

1. Боброва, И. И. Информатика : учебное пособие / И. И. Боброва ; МГТУ. - Магнитогорск : МГТУ, 2016. - 1 электрон. опт. диск (CD-ROM). - Загл. с титул. экрана. - URL: <https://magtu.informsystema.ru/uploader/fileUpload?name=2469.pdf&show=dcatalogues/1/1130212/2469.pdf&view=true> (дата обращения: 04.10.2019). - Макрообъект. - Текст : электронный. - Сведения доступны также на CD-ROM.

2. Лактионова, Ю. С. Информатика : учебное пособие / Ю. С. Лактионова, Л. С. Брябрина ; МГТУ. - Магнитогорск : МГТУ, 2015. - 1 электрон. опт. диск (CD-ROM). - Загл. с титул. экрана. - URL: <https://magtu.informsystema.ru/uploader/fileUpload?name=1507.pdf&show=dcatalogues/1/1124041/1507.pdf&view=true> (дата обращения: 04.10.2019). - Макрообъект. - Текст : электронный. - Сведения доступны также на CD-ROM.

### **б) Дополнительная литература:**

3. Романов, Е. П. Электронно-вычислительная техника и программирование : учебно-методическое пособие / Е. П. Романов ; МГТУ. - Магнитогорск : МГТУ, 2017. - 1 электрон. опт. диск (CD-ROM). - Загл. с титул. экрана. - URL: <https://magtu.informsystema.ru/uploader/fileUpload?name=3068.pdf&show=dcatalogues/1/1135237/3068.pdf&view=true> (дата обращения: 04.10.2019). - Макрообъект. - Текст : электронный. - Сведения доступны также на CD-ROM.

4. Самарина, И. Г. Программирование и основы алгоритмизации : учебное пособие. Ч. 1. Курс лекций / И. Г. Самарина. - Магнитогорск : МГТУ, 2013. - 1 электрон. опт. диск (CD-ROM). - Загл. с титул. экрана. - URL: <https://magtu.informsystema.ru/uploader/fileUpload?name=908.pdf&show=dcatalogues/1/1118881/908.pdf&view=true> (дата обращения: 04.10.2019). - Макрообъект. - Текст : электронный. - Сведения доступны также на CD-ROM.

### **в) методические указания**

5. Торшина, О. А. Объектно-ориентированное программирование : учебное пособие / О. А. Торшина ; МГТУ. - Магнитогорск : МГТУ, 2018. - 1 электрон. опт. диск (CD-ROM). - Загл. с титул. экрана. - URL: <https://magtu.informsystema.ru/uploader/fileUpload?name=3613.pdf&show=dcatalogues/1/1524595/3613.pdf&view=true> (дата обращения: 04.10.2019). - Макрообъект. - Текст : электронный. - ISBN 978-5-9967-1132-1. - Сведения доступны также на CD-ROM.

6. Методические указания по подготовке реферата представлены в приложении 1.

7. Методические указания по выполнению индивидуальных домашних заданий

представлены в приложении 2.

### г) Программное обеспечение и Интернет-ресурсы:

Наименование ПО	№ договора	Срок действия лицензии
MS Windows 7	Д-1227 от 08.10.2018	11.10.2021
MS Office 2007	№135 от 17.09.2007	бессрочно
FAR Manager	свободно распространяемое	бессрочно
7Zip	свободно распространяемое	бессрочно
AnyLogic University	Д-895-14 от 14.07.2014	бессрочно

1. Международная справочная система «Полпред» polpred.com. отрасль «Образование, наука». – URL: <http://education.polpred.com>.

2. Национальная информационно-аналитическая система. – Российский индекс научного цитирования (РИНЦ). – URL: [https://elibrary.ru/project\\_risc.asp](https://elibrary.ru/project_risc.asp).

3. Поисковая система Академия Google (Google Scholar). – URL: <https://scholar.google.ru>

4. Информационная система. – Единое окно доступа к информационным ресурсам. – URL: <http://window.edu.ru>

### 9 Материально-техническое обеспечение дисциплины (модуля)

Материально-техническое обеспечение дисциплины включает:

Тип и название аудитории	Оснащение аудитории
Учебные аудитории для проведения занятий лекционного типа	Мультимедийные средства хранения, передачи и представления информации
Учебные аудитории для проведения практических занятий, занятий семинарского типа, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации	Мультимедийные средства хранения, передачи и представления информации
Помещения для самостоятельной работы обучающихся	Персональные компьютеры с пакетом MS Office, выходом в Интернет и с доступом в электронную информационно-образовательную среду университета
Помещения для хранения и профилактического обслуживания учебного оборудования	Стеллажи для хранения учебно-методической документации, учебного оборудования и учебно-наглядных пособий

### Методические рекомендации по подготовке реферата

Реферат это продукт самостоятельной работы обучающегося, представляющий собой краткое изложение в письменном виде полученных результатов теоретического анализа определенной научной (учебно-исследовательской) темы, где автор раскрывает суть исследуемой проблемы, приводит различные точки зрения, а также собственные взгляды на неё.

Реферат – сбор и представление исчерпывающей информации по заданной теме из различных источников, приведение интересных фактов, статистических данных.

Работу над рефератом можно условно подразделить на три этапа:

1. Подготовительный этап, включающий изучение предмета исследования;
2. Изложение результатов изучения в виде связного текста;
3. Устное сообщение по теме реферата.

Текст реферата должен подчиняться определенным требованиям: он должен раскрывать тему, обладать связностью и цельностью.

Раскрытие темы предполагает, что в тексте реферата излагается относящийся к теме материал и предлагаются пути решения содержащейся в теме проблемы; связность текста предполагает смысловую соотносительность отдельных компонентов, а цельность - смысловую законченность текста.

С точки зрения связности все тексты делятся на тексты-констатации и тексты-рассуждения. Тексты-констатации содержат результаты ознакомления с предметом и фиксируют устойчивые и несомненные суждения. В текстах-рассуждениях одни мысли извлекаются из других, некоторые ставятся под сомнение, дается им оценка, выдвигаются различные предположения.

Структура реферата

- 1) титульный лист (оформляется по образцу, утвержденному кафедрой);
- 2) план работы с указанием страниц каждого пункта;
- 3) введение (обоснование актуальности выбранной для изучения темы для теории и практики, для автора реферата);
- 4) текстовое изложение материала по вопросам плана с необходимыми ссылками на источники, использованные автором реферата, с изложением собственной авторской позиции к обсуждаемой теме);
- 5) заключение;
- 6) список использованной литературы;
- 7) приложения, которые состоят из таблиц, фотографий, диаграмм, графиков, рисунков, схем (необязательная часть реферата).

Во введении аргументируется актуальность исследования, -

т. е. выявляется практическое и теоретическое значение данного исследования. Далее констатируется, что сделано в данной области предшественниками; перечисляются положения, которые должны быть обоснованы. Введение может также содержать обзор источников или экспериментальных данных, уточнение исходных понятий и терминов, сведения о методах исследования. Во введении обязательно формулируются цель и задачи реферата.

Объем введения - в среднем около 10% от общего объема реферата.

Основная часть реферата раскрывает содержание темы. Она наиболее значительна по объему, наиболее значима и ответственна. В ней обосновываются основные тезисы реферата, приводятся развернутые аргументы, предполагаются гипотезы, касающиеся существа обсуждаемого вопроса. Важно проследить, чтобы основная часть не имела форму монолога. Аргументируя собственную позицию, можно и должно анализировать и оценивать позиции различных исследователей, с чем-то соглашаться, чему-то возражать, кого-то опровергать. Текст основной части делится на главы, параграфы, пункты. План основной части может быть составлен с использованием различных методов группировки мате-

риала: классификации (эмпирические исследования), типологии (теоретические исследования), периодизации (исторические исследования).

Заключение — последняя часть научного текста. В ней краткой и сжатой форме излагаются полученные результаты, представляющие собой ответ на главный вопрос исследования. Здесь же могут намечаться и дальнейшие перспективы развития темы. Небольшое по объему сообщение также не может обойтись без заключительной части - пусть это будут две-три фразы. Но в них должен подводиться итог проделанной работы.

Реферат любого уровня сложности обязательно сопровождается списком используемой литературы. Названия книг в списке располагают по алфавиту с указанием выходных данных использованных книг.

Шкала оценивания

2 балла – тема не раскрыта на теоретическом уровне;

3 балл - тема раскрыта на теоретическом уровне;

4 баллов - тема раскрыта, студент свободно ориентируется в материале, приводит практические примеры;

5 баллов - тема раскрыта, студент свободно ориентируется в материале, приводит практические примеры, отвечает на вопросы группы и преподавателя, защиту сопровождает презентация.

## **Приложение 2 - Методические указания для выполнения домашних индивидуальных заданий**

Методические указания по выполнению индивидуальных домашних заданий представлены на образовательном портале МГТУ: [newlms.mgtu.ru](http://newlms.mgtu.ru)