



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет им. Г.И. Носова»



УТВЕРЖДАЮ
Директор ИГДиТ
И.А. Пыталев

15.03.2021 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

ОСНОВЫ ИНФОРМАТИКИ В ТРАНСПОРТНЫХ ПРИЛОЖЕНИЯХ

Направление подготовки (специальность)
38.03.02 Менеджмент

Направленность (профиль/специализация) программы
Логистика

Уровень высшего образования - бакалавриат

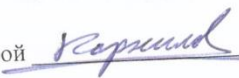
Форма обучения
очная

Институт/ факультет	Институт горного дела и транспорта
Кафедра	Логистика и управление транспортными системами
Курс	1
Семестр	2

Магнитогорск
2021 год

Рабочая программа составлена на основе ФГОС ВО - бакалавриат по направлению подготовки 38.03.02 Менеджмент (приказ Минобрнауки России от 12.08.2020 г. № 970)

Рабочая программа рассмотрена и одобрена на заседании кафедры Логистика и управление транспортными системами
04.02.2021, протокол № 6

Зав. кафедрой  С.Н. Корнилов

Рабочая программа одобрена методической комиссией ИГДиТ
15.03.2021 г. протокол № 5

Председатель  И.А. Пыталев

Рабочая программа составлена:
доцент кафедры ЛиУТС, канд. техн. наук

 П.Н. Мишуров

Рецензент:

Ведущий инженер-технолог ПГТ УЛ ПАО "ММК"  Е.В. Полежаев

Лист актуализации рабочей программы

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2022 - 2023 учебном году на заседании кафедры Логистика и управление транспортными системами

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ С.Н. Корнилов

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2023 - 2024 учебном году на заседании кафедры Логистика и управление транспортными системами

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ С.Н. Корнилов

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2024 - 2025 учебном году на заседании кафедры Логистика и управление транспортными системами

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ С.Н. Корнилов

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2025 - 2026 учебном году на заседании кафедры Логистика и управление транспортными системами

Протокол от _____ 20__ г. № ____
Зав. кафедрой _____ С.Н. Корнилов

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

Дисциплина Основы информатики в транспортных приложениях входит в часть учебного плана формируемую участниками образовательных отношений образовательной программы.

Для изучения дисциплины необходимы знания (умения, владения), сформированные в результате изучения дисциплин/ практик:

Основы логистики и управление цепями поставок

Математика

Знания (умения, владения), полученные при изучении данной дисциплины будут необходимы для изучения дисциплин/практик:

Системный анализ в логистике

Информационные системы в логистике

Имитационное 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:

Код индикатора	Индикатор достижения компетенции
ПК-1	Владением навыками количественного и качественного анализа информации при принятии управленческих решений, построения экономических, финансовых и организационно-управленческих моделей путем их адаптации к конкретным задачам управления
ПК-1.1	Находит в базах данных нужную информацию
ПК-1.2	Разрабатывает математические модели логистических систем
ПК-1.3	Выявляет конкретные пути повышения качества транспортного обслуживания клиентов, развития инфраструктуры товарного рынка и каналов распределения

4. Структура, объём и содержание дисциплины (модуля)

Общая трудоемкость дисциплины составляет 4 зачетных единиц 144 акад. часов, в том числе:

- контактная работа – 73,9 акад. часов;
- аудиторная – 72 акад. часов;
- внеаудиторная – 1,9 акад. часов;
- самостоятельная работа – 70,1 акад. часов;
- в форме практической подготовки – 0 акад. час;

Форма аттестации - зачет

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа студента	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код компетенции
		Лек.	лаб. зан.	практ. зан.				
1. Раздел «Программное обеспечение и вычислительной техники и его классификация»								
1.1 «Основные понятия теории информации»	2	2		2	2	Поиск дополнительной информации по заданной теме	Устный опрос	ПК-1.1, ПК-1.2, ПК-1.3
1.2 Тема «Тенденции развития программного обеспечения для ЭВМ и систем коммуникации»		2		2	2	Поиск дополнительной информации по заданной теме	Устный опрос	ПК-1.1, ПК-1.2, ПК-1.3
1.3 «Классификация базового и прикладного программного обеспечения»		2		2	2	Поиск дополнительной информации по заданной теме	Устный опрос	ПК-1.1, ПК-1.2, ПК-1.3
1.4 «Примеры использования программного обеспечения в различных отраслях народного хозяйства»		2		2	2	Самостоятельное изучение учебной и научной литературы	Устный опрос	ПК-1.1, ПК-1.2, ПК-1.3
Итого по разделу		8		8	8			
2. Раздел «Объектно-ориентированный подход к компьютерному программированию»								
2.1 «Основы объектно-ориентированный подхода к компьютерному программированию»	2	2		2	2	Самостоятельное изучение учебной и научной литературы	Устный опрос	ПК-1.1, ПК-1.2, ПК-1.3
2.2 «Типы данных Java и оператор if»		2		2	2	Подготовка к практическим занятиям, выполнение практических работ	Проверка практических заданий, устный опрос.	ПК-1.1, ПК-1.2, ПК-1.3
2.3 «Массивы»		2		2	2	Подготовка к практическим занятиям, выполнение практических работ	Проверка практических заданий, устный опрос	ПК-1.1, ПК-1.2, ПК-1.3

2.4 «Циклы»		2		2	2	Подготовка к практическим занятиям, выполнение практических работ	Проверка практических заданий, устный опрос.	ПК-1.1, ПК-1.2, ПК-1.3
2.5 «Коллекции»		2		2	2	Подготовка к практическим занятиям, выполнение практических работ	Проверка практических заданий, устный опрос.	ПК-1.1, ПК-1.2, ПК-1.3
2.6 «Операторы сравнения, логические операторы и switch»		2		2	2	Подготовка к практическим занятиям, выполнение практических работ	Подготовка к практическим занятиям, выполнение практических работ	ПК-1.1, ПК-1.2, ПК-1.3
2.7 «Видимость и связи»		2		2/2И	2	Подготовка к практическим занятиям, выполнение практических работ	Проверка практических заданий, устный опрос.	ПК-1.1, ПК-1.2, ПК-1.3
2.8 «Взаимодействие объектов в AnyLogic»		2		2/2И	2	Подготовка к практическим занятиям, выполнение практических работ	Проверка практических заданий, устный опрос.	ПК-1.1, ПК-1.2, ПК-1.3
2.9 «Диаграммы состояний»		2		2/2И	2	Подготовка к практическим занятиям, выполнение практических работ	Проверка практических заданий, устный опрос.	ПК-1.1, ПК-1.2, ПК-1.3
Итого по разделу		18		18/6И	18			
3. Раздел «Основные понятие метода имитационного моделирования»								
3.1 «Основные понятие метода имитационного моделирования»	2	2		2/2И	4	Поиск дополнительной информации по заданной теме	Устный опрос	ПК-1.1, ПК-1.2, ПК-1.3
3.2 «Системно-динамический подход к построению имитационных моделей»		2		2/2И	10	Поиск дополнительной информации по заданной теме	Устный опрос	ПК-1.1, ПК-1.2, ПК-1.3
3.3 «Дискретно-событийный и агентный подходы к построению имитационных моделей»		2		2/2И	10	Поиск дополнительной информации по заданной теме	Устный опрос	ПК-1.1, ПК-1.2, ПК-1.3
3.4 «Особенности построения имитационных моделей транспортных систем»		2		2/2И	10	Самостоятельное изучение учебной и научной литературы	Устный опрос	ПК-1.1, ПК-1.2, ПК-1.3
3.5 «Универсальные системы построения имитационных моделей. Инструмент имитационного моделирования AnyLogic»		2		2/0,4И	10,1	Поиск дополнительной информации по заданной теме	Устный опрос	ПК-1.1, ПК-1.2, ПК-1.3
Итого по разделу		10		10/8,4И	44,1			
Итого за семестр		36		36/14,4И	70,1		зачёт	

Итого по дисциплине	36		36/14,4 И	70,1		зачет	
---------------------	----	--	--------------	------	--	-------	--

5 Образовательные 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

Представлено в приложении 1.

7 Оценочные средства для проведения промежуточной аттестации

Представлены в приложении 2.

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:

1. 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.

2. 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:

1. 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.

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

3. Методические указания по выполнению индивидуальных домашних заданий представлены в приложении 2.

г) Программное обеспечение и Интернет-ресурсы:

Программное обеспечение

Наименование ПО	№ договора	Срок действия лицензии
MS Windows 7 Professional(для классов)	Д-1227-18 от 08.10.2018	11.10.2021
MS Windows 7 Professional (для классов)	Д-757-17 от 27.06.2017	27.07.2018
MS Office 2007 Professional	№ 135 от 17.09.2007	бессрочно
7Zip	свободно распространяемое	бессрочно
FAR Manager	свободно распространяемое	бессрочно

Профессиональные базы данных и информационные справочные системы

Название курса	Ссылка
Поисковая система Академия Google (Google Scholar)	URL: https://scholar.google.ru/
Национальная информационно-аналитическая система – Российский индекс научного цитирования (РИНЦ)	URL: https://elibrary.ru/project_risc.asp

9 Материально-техническое обеспечение дисциплины (модуля)

Материально-техническое обеспечение дисциплины включает:

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.

Приложение 1

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

```
int largerNum;
int lowNum = 9;
int highNum = 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 = newString[] { "Света", "Таня", "Маша" };

```

Двумерный массив значений типа double

```

double[][] двумерныйМассив = newdouble[10][20];
intчислоСтрок = двумерныйМассив.length;
intчислоСтолбцов = 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»

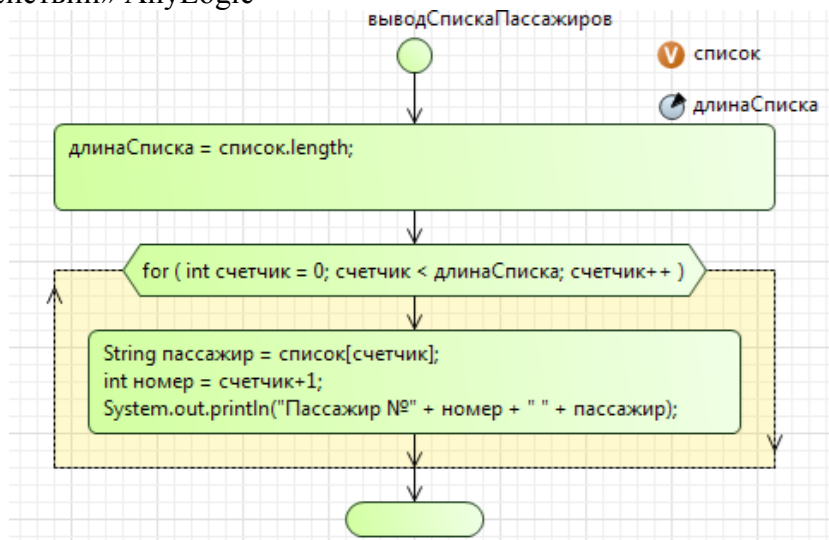
Пример цикла `for` на языке Java:

```
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 определенных 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.

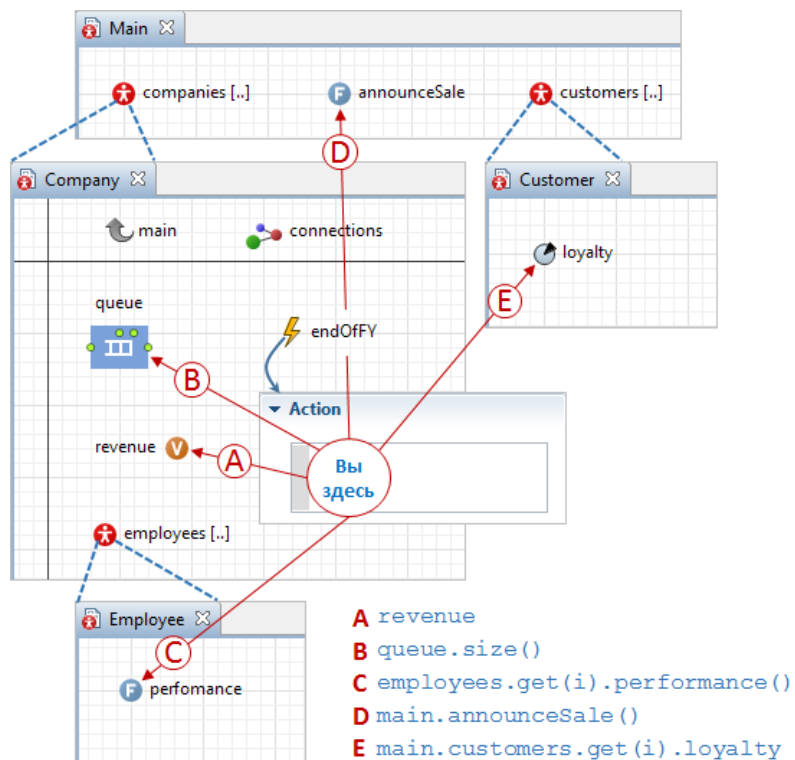
Создать коллекцию `LinkedList`, инициализировать ее тридцатью элементами. Идентифицировать десятый и двадцатый элемент коллекции. Добавить в коллекцию 5 элементов. Удалить первый элемент коллекции.

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 {}
```

Приложение 2

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
ПК-1: Владением навыками количественного и качественного анализа информации при принятии управленческих решений, построения экономических, финансовых и организационно-управленческих моделей путем их адаптации к конкретным задачам управления		
ПК-1.1	Находит в базах данных нужную информацию	Перечень теоретических вопросов: 1. Понятие об имитационном моделировании. Назначение имитационных моделей. 2. Современные подходы к имитационному моделированию. Сущность системно-динамического, дискретно-событийного (процессного) и агентного подходов к имитационному моделированию. 3. Область применения системно-динамических имитационных моделей. 4. Область применения дискретно-событийных (процессных) имитационных моделей. 5. Область применения агентных имитационных моделей. 6. Программные системы построения имитационных моделей. Их достоинства, недостатки и области применения. Достоинства мультиподходной системы AnyLogic. 7. Особенности системы построения имитационных моделей AnyLogic. Рабочая область системы. Состав рабочей области программы. Палитры блоков. 8. Сущность объектно-ориентированного подхода к разработке компьютерных программ. 9. Понятие «абстракция» в объектно-ориентированном программировании. Примеры. 10. Понятие «инкапсуляция» в объектно-ориентированном программировании. Примеры. 11. Понятие «наследование» в объектно-ориентированном программировании. Примеры. 12. Понятие «полиморфизм» в объектно-ориентированном программировании. Примеры.
ПК-1.2	Разрабатывает математические модели логистических систем	Примерные практические задания: Практическое задание №1 на тему «Типы данных» Общие правила приведения типов переменных Java в выражениях: <ul style="list-style-type: none">• типы переменных byte, short и char повышаются до типа int;• разрядность целевой переменной (выражения) соответствует максимальной

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		<p>разрядно-сти операнда (переменной выражения). Пример корректного использования типа целевой переменной: 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);</p> <p style="text-align: center;">Практическое задание №2 на тему «Логические операторы»</p> <p>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)</p> <p>Конструкция if-else intlargerNum; intlowNum = 9; inthighNum = 27;</p> <p>if(lowNum<highNum) // если первое число меньше второго { largerNum = highNum; } else { // иначе</p>

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		<pre> 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. 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; </pre>

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		<pre>break; case"прочие":код = 9; break; default: System.out.println("Неверныйтипвагона"); break; } if (код != 0) System.out.println("Типвагона: " + вагон + ",код: " + код);</pre>
ПК-1.3	Выявляет конкретные пути повышения качества транспортного обслуживания клиентов, развития инфраструктуры товарного рынка и каналов распределения	<p>Примерные практические задания: Практическое задание №3 на тему «Массивы» Пример одновременного создания и инициализации массива в AnyLogic (запись помещается в поле «Начальное значение» переменной) <pre>newint[] { 13, x-3, -15, 0, max(a,100) };</pre> или в программе на языкеJava <pre>int[] intarray = new int[] { 13, x-3, -15, 0, max(a, 100) };</pre> Пример создания массива строковых переменных <pre>String[] список = newString[] { “Света”, “Таня”, “Маша” };</pre> Двумерный массив значений типа double <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> При инициализации многомерных массивов используют дополнительные фигурные скобки, например, в программе на языке Java <pre>int[][] intarrayD = { { 1, 2, 3 }, { 4, 5, 6 } }</pre> </p>

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства															
		<p>или при указании начального значения переменной-массива в AnyLogic <code>newint[][] {{ 1, 2, 3 }, { 4, 5, 6 }}</code></p> <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 491 1789 647"> <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;">Практическое задание №4 на тему «Циклы»</p> <p>Пример цикла for на языке Java: <code>String[] список = new String[] { "Света", "Таня", "Маша", "Дима" }; intдлинаСписка = список.length; intсчетчик; for (счетчик = 0; счетчик < длинаСписка; счетчик++) { String пассажир = список[счетчик]; int номер = счетчик + 1; System.out.println("Пассажир №" + номер + " " + пассажир); }</code></p> <p>В результате выполнения цикла будут напечатано: Пассажир №1 Света Пассажир №2 Таня Пассажир №3 Маша Пассажир №4 Дима</p> <p>На следующем рисунке представлен пример создания цикла for при помощи «диаграммы действий» AnyLogic</p>	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
		<div data-bbox="1106 344 1930 887" data-label="Diagram"> <pre> graph TD Start(()) --> Process1[длинаСписка = список.length;] Process1 --> Loop{for (int счетчик = 0; счетчик < длинаСписка; счетчик++)} Loop --> Process2["String пассажир = список[счетчик]; int номер = счетчик+1; System.out.println("Пассажир №" + номер + " " + пассажир);"] Process2 --> Loop Loop --> End((())) </pre> </div> <p data-bbox="875 954 1032 983">Цикл while</p> <p data-bbox="875 1010 2159 1078">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 data-bbox="875 1086 1411 1238"> inti = 1; while (i < 5) { System.out.println(i + " "); i++; } </pre> <p data-bbox="875 1281 2159 1350">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</p> <pre data-bbox="875 1358 1411 1469"> inti = 1; while (i < 0) { System.out.println(i + " "); i++; } </pre>

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		<pre> } </pre> <p>Цикл while может выполняться бесконечное число раз, например</p> <pre> int i = 1; while (true) { System.out.println(i + " "); i++; } </pre> <p>Цикл do... while</p> <p>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.</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 < 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 < 5); </pre> <p style="text-align: center;">Практическое задание №5 на тему «Коллекции»</p> <p>Создать коллекцию <code>ArrayList</code>, инициализировать ее тридцатью элементами. Идентифицировать десятый и двадцатый элемент коллекции. Добавить в коллекцию 5 элементов. Удалить первый элемент коллекции и заменить одиннадцатый.</p> <p>Методы коллекции <code>ArrayList</code>:</p>

Структурный элемент компетенции	Планируемые результаты обучения	Оценочные средства
		<ul style="list-style-type: none"> – add(элемент)– добавление элемента в конец коллекции. Элемент может переменной любого типа Java, экземпляром класса Javaили 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 будет true, в противном случае – false. – remove(индекс)илиremove(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. <p>Создать коллекцию LinkedList, инициализировать ее тридцатью элементами. Идентифицировать десятый и двадцатый элемент коллекции. Добавить в коллекцию 5 элементов. Удалить первый элемент коллекции.</p> <p>Дополнительные методы коллекции LinkedList:</p> <ul style="list-style-type: none"> – 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.

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.