

Документ подписан простой электронной подписью

Информация о владельце:

ФИО: Цветлюк Лариса Сергеевна

Должность: Ректор

Дата подписания: 04.07.2025 14:33:55

Уникальный программный ключ:

e4e919f04dc802624637575c97796a744138b172b88dd38f9301d8c2340974f9

Автономная некоммерческая организация
высшего образования
«Институт непрерывного образования»

Рассмотрено
на заседании кафедры естественнонаучных
и общегуманитарных дисциплин
Зав. кафедрой



Трубичин А.С.
28 апреля 2025 г.

УТВЕРЖДАЮ:

Ректор АНО ВО «ИНО»



Цветлюк Л.С.
28 апреля 2025 г.

РАБОЧАЯ ПРОГРАММА УЧЕБНОЙ ДИСЦИПЛИНЫ
«Математическое моделирование и статистическая обработка данных»
для направления подготовки
08.04.01. «Строительство»,
Направленность (профиль) подготовки (программа)
«Технология и организация строительства»
(квалификация – магистр)

Москва, 2025

Рабочая программа учебной дисциплины «Математическое моделирование и статистическая обработка данных» разработана Годуновым Д.Г.

Рабочая программа учебной дисциплины «**Математическое моделирование и статистическая обработка данных**» разработана на основании федерального государственного образовательного стандарта высшего образования - магистратура по направлению подготовки **08.04.01 Строительство**, утвержденного приказом Министерства образования и науки Российской Федерации от 31 мая 2017 г. № 482, учебного плана по основной профессиональной образовательной программе высшего образования по направлению подготовки **08.04.01 Строительство**.

СОДЕРЖАНИЕ

1. Общие положения.....	4
1.1. Цель и задачи учебной дисциплины.....	4
1.2. Место учебной дисциплины в структуре основной профессиональной образовательной программы.....	4
1.3. Планируемые результаты обучения по учебной дисциплине в рамках планируемых результатов освоения основной профессиональной образовательной программы.....	4
2. Объем учебной дисциплины, включая контактную работу обучающегося с преподавателем и самостоятельную работу обучающегося.....	5
3. Содержание учебной дисциплины.....	6
3.1. Учебно-тематический план по очной форме обучения.....	65 6
3.2. Учебно-тематический план по очно-заочной форме обучения.....	7 7
3.3. Учебно-тематический план по заочной форме обучения.....	85 8
4. Учебно-методическое обеспечение самостоятельной работы обучающихся по учебной дисциплине.....	8
5. Фонд оценочных средств для проведения текущей и промежуточной аттестации обучающихся по учебной дисциплине.....	11
5.1. Форма промежуточной аттестации обучающегося по учебной дисциплине....	11
5.2. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы.....	11
5.3. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания.....	12
5.4. Типовые контрольные задания или иные материалы, необходимые для оценки знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций в процессе освоения образовательной программы..	14
5.5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций.....	60
6. Перечень основной и дополнительной литературы для освоения учебной дисциплины.....	60
6.1. Основная литература.....	60
6.2. Дополнительная литература.....	60
7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», современные профессиональные базы данных и информационные системы, необходимые для освоения учебной дисциплины.....	61
8. Методические указания для обучающихся по освоению учебной дисциплины.....	61
9. Программное обеспечение информационно-коммуникационных технологий	62
9.1. Информационные технологии.....	62
9.2. Программное обеспечение.....	62
9.3. Информационно-справочные системы.....	63
10. Перечень материально-технического обеспечения необходимого для реализации программы по учебной дисциплине.....	63
11. Образовательные технологии.....	63

1. Общие положения

1.1. Цель и задачи дисциплины

Целью дисциплины «Математическое моделирование и статистическая обработка данных» является формирование у обучающихся знаний, умений, навыков по теории и практике математического моделирования и статистической обработке данных вероятностных процессов для их применения при решении задач в будущей профессиональной деятельности..

Задачи дисциплины и профессиональной деятельности выпускников:

- приобрести основы теоретических знаний по теории статистического моделирования;
- научиться воспроизводить случайные величины заданных законов распределения;
- вычислять значения величин, характеризующих функционирование объектов моделирования;
- устанавливать связь алгоритмов моделирования с алгоритмами решения задач математического моделирования.

1.2. Место учебной дисциплины в структуре основной профессиональной образовательной программы.

Дисциплина относится к обязательной части Блока 1. Дисциплины (модули).

1.3. Планируемые результаты обучения по учебной дисциплине в рамках планируемых результатов освоения основной профессиональной образовательной программы.

Процесс освоения учебной дисциплины направлен на формирование у обучающихся следующих компетенций: УК-1, ОПК-1, ПК-2.

В процессе изучения дисциплины у обучающихся должны быть сформированы следующие компетенции:

Код компетенции	Содержание компетенции	Индикаторы достижения компетенций
УК-1	Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий	УК-1.1. Описание сути проблемной ситуации УК-1.2. Выявление составляющих проблемной ситуации и связей между ними УК-1.3. Сбор и систематизация информации по проблеме УК-1.4. Оценка адекватности и достоверности информации о проблемной ситуации УК-1.5. Выбор методов критического анализа, адекватных проблемной ситуации УК-1.6. Разработка и обоснование плана действий по решению проблемной ситуации УК-1.7. Выбор способа обоснования решения (индукция, дедукция, по аналогии) проблемной ситуации
ОПК-1	Способен решать задачи профессиональной деятельности на основе использования теоретических и практических основ, математического аппарата фундамен-	ОПК-1.1. Выбор фундаментальных законов, описывающих изучаемый процесс или явление ОПК-1.2. Составление математической модели, описывающей изучаемый процесс или явление, выбор и обоснование граничных и начальных условий

	тальных наук	ОПК-1.3. Оценка адекватности результатов моделирования, формулирование предложений по использованию математической модели для решения задач профессиональной деятельности ОПК-1.4. Применение типовых задач теории оптимизации в профессиональной деятельности
ПК-2	Способность осуществлять преподавательскую деятельность по программам профессионального обучения и образования в области жилищно-коммунального хозяйства	ПК-2.1. Постановка учебных целей в виде основных показателей достижения результата обучения ПК-2.2. Составление плана-конспекта проведения учебного занятия ПК-2.3. Выбор учебных заданий, адекватных учебной цели ПК-2.4. Выбор формы групповой работы и образовательной технологии при проведении практического занятия ПК-2.5. Выбор методов обучения, адекватных учебной цели ПК-2.6. Контроль и оценка освоения обучающимися учебного материала

2. Объем учебной дисциплины, включая контактную работы обучающегося с преподавателем и самостоятельную работу обучающегося

Общая трудоемкость учебной дисциплины составляет **6** зачетных единиц.

Очная форма обучения

Вид учебной работы	Всего часов	Семестр
		1
Аудиторные учебные занятия, всего	80	80
В том числе:		
Учебные занятия лекционного типа	32	32
Практические занятия	32	32
Лабораторная работа	16	16
Самостоятельная работа обучающихся, всего	127	127
В том числе:		
Самоподготовка	80	80
Доклады/рефераты	25	25
Тестирование	22	22
Контроль: вид промежуточной аттестации (экзамен)	9	Экзамен
Общая трудоемкость учебной дисциплины	216	216

Очно-заочная форма обучения

Вид учебной работы	Всего часов	Семестры	
		1	2
Аудиторные учебные занятия, всего	80	32	48
В том числе:			
Учебные занятия лекционного типа	32	16	16
Практические занятия	32	16	16
Лабораторная работа	16	-	16
Самостоятельная работа обучающихся,	123	72	51

всего			
В том числе:			
Самоподготовка	80	40	40
Доклады/рефераты	23	13	10
Тестирование	20	10	10
Контроль: вид промежуточной аттестации (экзамен)	13	Зачет 4	Экзамен 9
Общая трудоемкость учебной дисциплины	216	108	108

Заочная форма обучения

Вид учебной работы	Всего часов	Семестры	
		1	2
Аудиторные учебные занятия, всего	26	10	16
В том числе:			
Учебные занятия лекционного типа	12	6	6
Практические занятия	8	4	4
Лабораторная работа	6	-	6
Самостоятельная работа обучающихся, всего	177	94	83
В том числе:			
Самоподготовка	100	50	50
Доклады/рефераты	42	22	20
Тестирование	35	20	15
Контроль: вид промежуточной аттестации (экзамен)	13	Зачет 4	Экзамен 9
Общая трудоемкость учебной дисциплины	216	108	108

3. Содержание учебной дисциплины

3.1. Учебно-тематический план по очной форме обучения

Объем аудиторных занятий составляет 80 ч.

Объем самостоятельной работы – 123ч.

№ п/п	Модуль, раздел (тема)	Виды учебной работы, академических часов						Формы контроля освоения обучающимися учебной дисциплины, рейтинговых баллов		
		Всего	Самостоятельная работа обучающихся	Контактная работа преподавателя с обучающимися				Текущий контроль освоения обучающимися учебной дисциплины		Промежуточная аттестация обучающихся
				Всего	Лекционного типа	Лабораторные занятия	Практические занятия	Тестирование	Рефераты/доклады	
1	Тема 1. Основные понятия теории вероятностей и математической статистики	41	29	12	8		4	+	+	

2	Тема 2. Математическая статистика	41	23	18	8	4	6	+	+	
3	Тема 3. Основные понятия математического моделирования	42	26	16	6	4	6	+	+	
4	Тема 4. Выборочное наблюдение и обработка результатов эксперимента. Проверка однородности выборки	42	24	18	6	4	8	+	+	
5	Тема 5 Анализ, интерпретация и оптимизация регрессионных моделей	42	26	16	4	4	8	+	+	
ВСЕГО ЧАСОВ		216	127	80	32	16	32			9

3.2. Учебно-тематический план по очно-заочной форме обучения

Объем аудиторных занятий составляет 26 ч.

Объем самостоятельной работы – 177 ч.

№ п/ п	Модуль, раздел (тема)	Виды учебной работы, академических часов						Формы контроля освоения обучающимися учебной дисциплины, рейтинговых баллов		
		Всего	Самостоятельная работа обучающегося	Контактная работа преподавателя с обучающимися				Текущий контроль освоения обучающимися учебной дисциплины		Промежуточная аттестация обучающихся
				Всего	Лекционного типа	Лабораторные занятия	Практические занятия	Тестирование	Рефераты/доклады	
1 семестр										
1	Тема 1. Основные понятия теории вероятностей и математической статистики	34	26	8	6		2		+	
2	Тема 2. Математическая статистика	35	23	12	5		7			
3	Тема 3. Основные понятия математического моделирования	35	23	12	5		7			
	Итого по семестру 1	108	72	32	16		16			4
2 семестр										
4	Тема 4. Выборочное наблюдение и обработка результатов эксперимента. Проверка однородности выборки	49	25	24	8	8	8			

5	Тема 5 Анализ, интерпретация и оптимизация регрессионных моделей	50	26	24	8	8	8			
	Итого по семестру 2	99	51	48	16	16	16			9
ВСЕГО ЧАСОВ		216	123	80	32	16	32			13

3.3. Учебно-тематический план по заочной форме обучения

Объем аудиторных занятий составляет 80 ч.

Объем самостоятельной работы – 127 ч.

№ п/ п	Модуль, раздел (тема)	Виды учебной работы, академических часов						Формы контроля освоения обучающимися учебной дисциплины, рейтинговых баллов		
		Всего	Самостоятельная работа обучающегося	Контактная работа преподавателя с обучающимися				Текущий контроль освоения обучающимися учебной дисциплины		Промежуточная аттестация обучающихся
				Всего	Лекционного типа	Лабораторные занятия	Практические занятия	Тестирование	Рефераты/доклады	
1семестр										
1	Тема 1. Основные понятия теории вероятностей и математической статистики	34	32	2	2				+	
2	Тема 2. Математическая статистика	35	31	4	2		2			
3	Тема 3. Основные понятия математического моделирования	35	31	4	2		2			
	Итого по семестру 1	108	94	10	6		4			4
2семестр										
4	Тема 4. Выборочное наблюдение и обработка результатов эксперимента. Проверка однородности выборки	49	41	8	3	3	2			
5	Тема 5 Анализ, интерпретация и оптимизация регрессионных моделей	50	42	8	3	3	2			
	Итого по семестру 2	99	83	16	6	6	4			9
ВСЕГО ЧАСОВ		216	123	80	12	6	8			13

4. Учебно-методическое обеспечение самостоятельной работы обучающихся по учебной дисциплине

Тема 1. Основные понятия теории вероятностей и математической статистики

Независимые испытания. Формула полной вероятности. Случайные величины. Математическое ожидание случайных величин. Независимость случайных величин. Дисперсия случайных величин. Среднеквадратичное отклонение. Распределение случайной величины: определение понятия. Стандартное нормальное распределение. Правило трех сигм. Центральная предельная теорема. Экспоненциальное распределение. Распределения. Вейбулла-Гнеденко. Дискретные распределения: биномиальные, гипергеометрические и Пуассона.

Вопросы для самоподготовки

1. Независимые испытания.
2. Случайные величины.
3. Стандартное нормальное распределение.
4. Дискретные распределения

Формы контроля самостоятельной работы обучающихся: проверка ответов на вопросы самоподготовки.

Тема 2. Математическая статистика

Статистические методы обработки экспериментальных данных.

Статистические оценки. Несмещенные, эффективные и состоятельные оценки. Выборочная средняя и выборочная дисперсия. Число степеней свободы. Основные законы распределения статистических оценок.

Точечная и интервальная оценки. Доверительный интервал. Метод моментов и метод наибольшего правдоподобия для точечной оценки параметров распределения.

Статистическая гипотеза. Проверка статистических гипотез. Ошибки первого и второго рода. Статистический критерий проверки нулевой гипотезы.

Проверка гипотезы о распределении генеральной совокупности. Критерий Пирсона.

Зависимости между случайными величинами. Типы зависимостей. Линейная связь. Корреляция. Регрессионный анализ. Выборочное уравнение регрессии. Отыскание параметров выборочного уравнения регрессии по несгруппированным и сгруппированным данным.

Дисперсионный анализ. Понятие о дисперсионном анализе. Факторная и остаточная дисперсии.

Основные понятия многомерного статистического анализа. Методы факторного анализа, их область применения. Метод главных компонент. Классификация объектов, описываемых количественными и качественными признаками. Примеры кластер-анализа.

Вопросы для самоподготовки

1. Статистические методы обработки экспериментальных данных.
2. Основные законы распределения статистических оценок.
3. Статистическая гипотеза.
4. Зависимости между случайными величинами.
5. Дисперсионный анализ.
6. Основные понятия многомерного статистического анализа

Формы контроля самостоятельной работы обучающихся: проверка ответов на вопросы самоподготовки.

Тема 3. Основные понятия математического моделирования.

Понятие модели. Классификация моделей. Понятие моделирования. Виды моделирования. Этапы моделирования. Назначение этапов моделирования. Стадии исследования.

Эксперимент. Виды экспериментов. Основные задачи, решаемые планированием эксперимента.

Вопросы для самоподготовки

1. Понятие и классификация моделей.
2. Моделирование: понятие, виды, этапы.
3. Эксперимент: виды, основные задачи

Формы контроля самостоятельной работы обучающихся: проверка ответов на вопросы самоподготовки.

Тема 4. Выборочное наблюдение и обработка результатов эксперимента. Проверка однородности выборки

Измерения. Виды измерений. Класс точности измерительных приборов. Абсолютная и относительная погрешность. Ошибки наблюдения и их виды (систематические, случайные, грубые). Выбросы. Правила округления чисел. Значащие цифры числа. Доверительное оценивание: доверительная вероятность, доверительный интервал, надежность. Взвешенное среднее. Обработка результатов прямых измерений. Обработка результатов косвенных измерений. Выборочная совокупность. Репрезентативность выборки. Виды выборок. Дисперсия. Распределение вероятностей Стьюдента. Проверка гипотез: ошибки первого, второго рода. Проверка однородности независимых выборок. Проверка воспроизводимости опытов. Критерий Стьюдента, критерий Фишера, критерий Кохрена. Выявление промахов при обработке результатов измерений. Критерий Шовене.

Вопросы для самоподготовки

1. Измерения: виды и класс точностей.
2. Обработка результатов прямых и косвенных измерений.
3. Проверка гипотез.
4. Выявление промахов при обработке результатов измерений

Формы контроля самостоятельной работы обучающихся: проверка ответов на вопросы самоподготовки.

Тема 5. Анализ, интерпретация и оптимизация регрессионных моделей.

Понятие регрессионной зависимости. Уравнение регрессии. Понятие корреляции. Основная задача регрессионного анализа. Основная задача корреляционного анализа. Однофакторная линейная регрессия. Определение коэффициентов регрессии линейной однофакторной зависимости методом наименьших квадратов. Выборочный коэффициент корреляции. Свойства коэффициента корреляции. Оценка значимости коэффициента корреляции. Множественная линейная регрессия. Полиномиальная регрессия. Проверка регрессии на достоверность и адекватность. Дисперсия воспроизводимости. Анализ остатков. Выделение факторов, оказывающих наибольшее влияние на отклик. Оценка характера, знака и степени влияния факторов на отклик. Прогноз значения отклика для любых значений факторов. Задача оптимизации. Метод крутого восхождения по поверхности отклика.

Вопросы для самоподготовки

1. Понятие регрессионной зависимости.
2. Понятие корреляции.
3. Проверка регрессии на достоверность и адекватность.
4. Выделение факторов, оказывающих наибольшее влияние на отклик.
5. Задача оптимизации.

Формы контроля самостоятельной работы обучающихся: проверка ответов на вопросы самоподготовки.

5. Фонд оценочных средств для проведения текущей и промежуточной аттестации обучающихся по учебной дисциплине

5.1. Форма промежуточной аттестации обучающегося по учебной дисциплине.

Контрольным мероприятием промежуточной аттестации обучающихся по учебной дисциплине является зачет (1 семестр для очно-заочной и заочной форм обучения), экзамен (1 семестр для очной формы обучения и 2 семестр для очно-заочной и заочной формы обучения), который проводится в устной форме.

5.2. Перечень компетенций с указанием этапов их формирования в процессе освоения образовательной программы.

Код компетенции	Содержание компетенции	Результаты обучения	Индикаторы достижения компетенций	Результаты обучения
УК-1	Способен осуществлять критический анализ проблемных ситуаций на основе системного подхода, вырабатывать стратегию действий	компоненты компетенции соотносятся с содержанием дисциплины, компетенция реализуется частично	УК-1.1. Описание сути проблемной ситуации УК-1.2. Выявление составляющих проблемной ситуации и связей между ними УК-1.3. Сбор и систематизация информации по проблеме УК-1.4. Оценка адекватности и достоверности информации о проблемной ситуации УК-1.5. Выбор методов критического анализа, адекватных проблемной ситуации УК-1.6. Разработка и обоснование плана действий по решению проблемной ситуации УК-1.7. Выбор способа обоснования решения (индукция, дедукция, по аналогии) проблемной ситуации	Знать: основные категории и понятия математического моделирования и статистической обработки данных Уметь: осуществлять критический анализ проблемных ситуаций на основе системного подхода Владеть: навыками выработки стратегии действий

ОПК-1	Способен решать задачи профессиональной деятельности на основе использования теоретических и практических основ, математического аппарата фундаментальных наук		ОПК-1.1. Выбор фундаментальных законов, описывающих изучаемый процесс или явление ОПК-1.2. Составление математической модели, описывающей изучаемый процесс или явление, выбор и обоснование граничных и начальных условий ОПК-1.3. Оценка адекватности результатов моделирования, формулирование предложений по использованию математической модели для решения задач профессиональной деятельности ОПК-1.4. Применение типовых задач теории оптимизации в профессиональной деятельности	Знать: основы математического аппарата фундаментальных наук Уметь: решать задачи профессиональной деятельности Владеть: навыками использования теоретических и практических основ, математического аппарата фундаментальных наук
ПК-2	Способность осуществлять преподавательскую деятельность по программам профессионального обучения и образования в области жилищно-коммунального хозяйства	компоненты компетенции соотносятся с содержанием дисциплины, компетенция реализуется частично	ПК-2.1. Постановка учебных целей в виде основных показателей достижения результата обучения ПК-2.2. Составление плана-конспекта проведения учебного занятия ПК-2.3. Выбор учебных заданий, адекватных учебной цели ПК-2.4. Выбор формы групповой работы и образовательной технологии при проведении практического занятия ПК-2.5. Выбор методов обучения, адекватных учебной цели ПК-2.6. Контроль и оценка освоения обучающимися учебного материала	Знать: основные методы профессионального обучения и образования Уметь: осуществлять преподавательскую деятельность по программам профессионального обучения Владеть: навыками профессионального обучения и образования

5.3. Описание показателей и критериев оценивания компетенций на различных этапах их формирования, описание шкал оценивания

Код компетенции	Этапы формирования компетенции	Инструмент, оценивающий сформированность компетенции*	Показатель оценивания компетенции
УК-1 ОПК-1 ПК-2	Этап формирования знаковой основы компетенций (этап формирования содержательно-теоретического базиса компетенции) Лекционные и практические занятия по темам Тема 1. Введение в курс «Деловые коммуникации» Тема 2. Типология деловых коммуникации Тема 3. Технологии формирования эффективных деловых коммуникаций Тема 4. Роль личностной эффективности в деловой коммуникации Тема 5. Воздействие и влияние в деловом общении.	Реферат*/Доклад* Тестирование* Зачет с оценкой*	А) полностью сформирована - 5 баллов Б) частично сформирована - 3-4 балла С) не сформирована - 2 и менее баллов

***Характеристики инструментов (средств), оценивающих сформированность компетенции:**

Реферат – продукт самостоятельной работы студента, 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 нее. В реферате должна быть раскрыта тема, структура должна соответствовать теме и быть отражена в оглавлении, размер работы – 10-15 стр. печатного текста (список литературы и приложения в объем не входят), снабженного сносками и списком использованной литературы. Текстовая часть работы состоит из введения, основной части и заключения. Во введении обучающийся кратко обосновывает актуальность избранной темы реферата, раскрывает цель и задачи, которые он собирается решить в ходе своего небольшого исследования. В основной части (может состоять из 2-3 параграфов) подробно раскрывается содержание вопросов темы. В заключении должны быть кратко сформулированы полученные результаты исследования, приведены обобщающие выводы. Заключение может включать предложения автора, в том числе и по дальнейшему изучению заинтересовавшей его проблемы. В список литературы обучающийся включает только те издания, которые он использовал при написании реферата (не менее 5-7). В тексте обязательны ссылки на использованную литературу, оформленные в соответствии с ГОСТом. В приложении к реферату могут выноситься таблицы, графики, схемы и другие вспомогательные материалы, на которые имеются ссылки в тексте реферата. **Критерии оценки реферата:** 1) Степень раскрытия сущности вопроса: а) соответствие плана теме реферата; б) соответствие содержания теме и плану реферата; в) полнота проанализированного материала по теме; умение работать с отечественными и зарубежными научными исследованиями, критической литературой, периодикой, систематизировать и структурировать материал; г) обоснованность способов и методов работы с материалом, адекватное и правомерное использование методов классификации, сравнения и др.; е) умение обобщать, делать выводы, сопоставлять различные точки зрения по одному вопросу (проблеме). 2) Оригинальность текста: а) самостоятельность в постановке проблемы, формулирование нового аспекта известной проблемы в установлении новых связей (межпредметных, внутрипредметных, ин-

теграционных); б) явленность авторской позиции, самостоятельность оценок и суждений; д) стилевое единство текста, единство жанровых черт. 3) Обоснованность выбора источников: а) оценка использованной литературы: привлечены ли наиболее известные работы по теме исследования (в т.ч. журнальные публикации последних лет, последние статистические данные, сводки, справки и т.д.). 4) Соблюдение требований к оформлению: а) насколько верно оформлены ссылки на используемую литературу, список литературы. б) оценка грамотности и культуры изложения (в т.ч. орфографической, пунктуационной, стилистической культуры), владение терминологией; в) соблюдение требований к объёму реферата.

Доклад – продукт самостоятельной работы обучающегося, представляющий собой публичное выступление по представлению полученных результатов решения определенной учебно-практической, учебно-исследовательской или научной темы. Доклад – это научное сообщение на практическом занятии, заседании научного кружка или учебно-теоретической конференции. **Критерии оценки доклада:** соответствие содержания заявленной теме; актуальность, новизна и значимость темы; аргументированность, полнота, структурированность и логичность изложения; свободное владение материалом: последовательность, умение вести дискуссию, правильно отвечать на вопросы; самостоятельность, степень оригинальности предложенных решений, иллюстративности, обобщений и выводов; наличие собственного отношения автора к рассматриваемой проблеме/теме (насколько точно и аргументировано выражено отношение автора к теме доклада); представление материала: качество презентации, оформления; культура речи, ораторское мастерство (соблюдение норм литературного языка, правильное произношение слов и фраз, оптимальный темп речи; умение правильно расставлять акценты; умение говорить достаточно громко, четко и убедительно); использование профессиональной терминологии (оценка того, насколько полно отражены в выступлении обучающегося профессиональные термины и общекультурные понятия по теме, а также насколько уверенно выступающий ими владеет); выдержанность регламента.

Тестирование – это контрольное мероприятие по учебному материалу, состоящее в выполнении обучающимся системы стандартизированных заданий, которая позволяет автоматизировать процедуру измерения уровня знаний и умений обучающегося. Тестирование включает в себя следующие типы заданий: задание с единственным выбором ответа из предложенных вариантов, задание на определение верных и неверных суждений; задание с множественным выбором ответов. **Критерии оценки:** от 90% до 100% правильно выполненных заданий – отлично; от 70% до 89% правильно выполненных заданий – хорошо; от 50% до 69% правильно выполненных заданий – удовлетворительно; от 0 до 49 % правильно выполненных заданий – не удовлетворительно.

Зачет с оценкой – контрольное мероприятие, которое проводится по учебной дисциплине в виде, предусмотренном учебным планом, по окончании изучения курса. Занятие аудиторное, проводится в устной форме с использованием фондов оценочных средств по учебной дисциплине.

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

Текущая аттестация.

Темы докладов /рефератов.

1. Роль и место моделирования в создании и исследовании систем.
2. Критерии качества математических моделей.
3. Основы математического моделирования: требования к моделям, свойства моделей, составление моделей, примеры.

4. Классификация методов построения моделей систем.
5. Построение моделей идентификации поисковыми методами.
6. Оценка точности и достоверности результатов моделирования.
7. Технология построения моделей (в общем случае и для конкретных схем).
8. Математическое моделирование как наука и искусство.
9. Современные методы прогнозирования явлений и процессов.
10. Классификация языков и систем моделирования.
11. Методики вычислительного (компьютерного) эксперимента.
12. Перспективы развития компьютерного моделирования сложных систем.
13. Математические схемы вероятностных автоматов.
14. Сети массового обслуживания и их применение.
15. Типовые математические модели сетей массового обслуживания (открытых и замкнутых).
16. Качественные методы моделирования систем.
17. Системная динамика как методология и инструмент исследования сложных процессов.
18. Анализ сложных систем с помощью моделей клеточных автоматов.
19. Эволюционное моделирование и генетические алгоритмы.
20. Современные подходы имитационного моделирования.
21. Распределенные системы имитационного моделирования.
22. Способы управления временем в имитационном моделировании.
23. Использование онтологий в имитационном моделировании.
24. Методы интеллектуального анализа данных.
25. Методы прогнозирования на основе нечетких временных рядов.
26. Косвенные методы построения функций принадлежности нечетких множеств.
27. Методы нечеткого моделирования.
28. Нечеткие методы классификации.
29. Использование нечетких представлений при построении и анализе моделей идентификации.
30. Определение и классификация неопределенностей в задачах моделирования систем.
31. Моделирование и анализ распределенных информационных систем.
32. Модификация сетей Петри для моделирования систем специального вида.
33. Обобщения сетей Петри.
34. Вложенные сети Петри и моделирование распределенных систем.
35. Классификация нечетких сетей Петри.
36. Многоагентные модели исследования систем.
37. Математические модели онтологии предметных областей.
38. Моделирование систем на основе анализа размерностей и теории подобия.
39. Модели информационного поиска в массиве документов.
40. Способы автоматизированного извлечения знаний о предметной области из текстов электронных документов.
41. Предметно-ориентированные системы научной осведомленности.
42. Нечеткие запросы к базам данных.

Тестирование по учебной дисциплине «Математическое моделирование и статистическая обработка данных»

1. Закончите предложение: «Объект, который используется в качестве «заместителя», представителя другого объекта с определенной целью, называется ...»
 1. моделью;
 2. копией;
 3. предметом;
 4. оригиналом.
2. Закончите предложение: «Модель, по сравнению с объектом-оригиналом, содержит ...»
 1. меньше информации;
 2. столько же информации;
 3. больше информации.
3. Моделирование — это:
 1. процесс замены реального объекта (процесса, явления) моделью, отражающей его существенные признаки с точки зрения достижения конкретной цели;
 2. процесс демонстрации моделей одежды в салоне мод;
 3. процесс неформальной постановки конкретной задачи;
 4. процесс замены реального объекта (процесса, явления) другим материальным или идеальным объектом;
 5. процесс выявления существенных признаков рассматриваемого объекта.
4. Процесс построения модели, как правило, предполагает:
 1. описание всех свойств исследуемого объекта;
 2. выделение наиболее существенных с точки зрения решаемой задачи свойств объекта;
 3. выделение свойств объекта безотносительно к целям решаемой задачи;
 4. описание всех пространственно-временных характеристик изучаемого объекта;
 5. выделение не более трех существенных признаков объекта.
5. Математическая модель объекта — это:
 1. созданная из какого-либо материала модель, точно отражающая внешние признаки объекта-оригинала;
 2. описание в виде схемы внутренней структуры изучаемого объекта;
 3. совокупность данных, содержащих информацию о количественных характеристиках объекта и его поведения в виде таблицы;
 4. совокупность записанных на языке математики формул, отражающих те или иные свойства объекта-оригинала или его поведение;
 5. последовательность электрических сигналов.
6. К числу математических моделей относится:
 1. милицейский протокол;
 2. правила дорожного движения;
 3. формула нахождения корней квадратного уравнения;
 4. кулинарный рецепт;
 5. инструкция по сборке мебели.
7. К числу документов, представляющих собой информационную модель управления государством, можно отнести:
 1. Конституцию РФ;
 2. географическую карту России;
 3. Российский словарь политических терминов;
 4. схему Кремля;
 5. список депутатов государственной Думы.
8. Рисунки, карты, чертежи, диаграммы, схемы, графики представляют собой:
 1. табличные информационные модели;

2. математические модели;
 3. натурные модели;
 4. графические информационные модели;
 5. иерархические информационные модели.
- 9 . Описание глобальной компьютерной сети Интернет в виде системы взаимосвязанных следует рассматривать как:
1. натурную модель;
 2. табличную модель;
 3. графическую модель;
 4. математическую модель;
 5. сетевую модель.
10. В биологии классификация представителей животного мира представляет собой:
1. иерархическую модель;
 2. табличную модель;
 3. графическую модель;
 4. математическую модель;
 5. натурную модель.
11. Информационной моделью организации занятий в школе является:
1. свод правил поведения учащихся;
 2. список класса;
 3. расписание уроков;
 4. перечень учебников.
12. Отметьте пропущенное слово: «Географическая карта является примером ... модели»
1. образной
 2. знаковой
 3. смешанной
 4. натурной
13. Укажите пары объектов, о которых можно сказать, что они находятся в отношении «объект – модель»:
1. компьютер – процессор
 2. Новосибирск – город
 3. слякоть – насморк
 4. автомобиль – техническое описание автомобиля
 5. город – путеводитель по городу
14. Модель есть замещение изучаемого объекта другим объектом, который отражает:
1. все стороны данного объекта
 2. некоторые стороны данного объекта
 3. существенные стороны данного объекта
 4. несуществующие стороны данного объекта
15. Что является моделью объекта «яблоко»?
1. муляж;
 2. фрукт;
 3. варенье;
 4. компот.
16. Укажите примеры натуральных моделей:
1. физическая карта
 2. глобус
 3. график зависимости расстояния от времени
 4. макет здания
 5. схема узора для вязания крючком
 6. муляж яблока
 7. манекен

17. Укажите примеры образных информационных моделей:
1. рисунок
 2. фотография
 3. словесное описание
 4. формула
18. Закончите предложение: "Можно создавать и использовать ..."
1. разные модели объекта
 2. единственную модель объекта
 3. только натурную модель объекта
19. Отметьте пропущенное слово: "Словесное описание горного ландшафта является примером ... модели"
1. образной
 2. знаковой
 3. смешанной
 4. натурной
20. Расписание движение поездов может рассматриваться как пример:
1. натурной модели;
 2. табличной модели;
 3. графической модели;
 4. компьютерной модели;
 5. математической модели.
21. Модель объекта это...
- 1) предмет похожий на объект моделирования
 - 2) объект -заместитель, который учитывает свойства объекта, необходимые для достижения цели
 - 3) копия объекта
 - 4) шаблон, по которому можно произвести точную копию объекта
22. Основная функция модели это:
- 1) Получить информацию о моделируемом объекте
 - 2) Отобразить некоторые характеристические признаки объекта
 - 3) Получить информацию о моделируемом объекте или отобразить некоторые характеристические признаки объекта
 - 4) Воспроизвести физическую форму объекта
23. Математические модели относятся к классу...
- 1) Изобразительных моделей
 - 2) Прагматических моделей
 - 3) Познавательных моделей
 - 4) Символических моделей
24. Математической моделью объекта называют...
- 1) Описание объекта математическими средствами, позволяющее выводить суждение о некоторых его свойствах при помощи формальных процедур
 - 2) Любую символическую модель, содержащую математические символы
 - 3) Представление свойств объекта только в числовом виде
 - 4) Любую формализованную модель
25. Методами математического моделирования являются ...
- 1) Аналитический
 - 2) Числовой
 - 3) Аксиоматический и конструктивный
 - 4) Имитационный
26. Какая форма математической модели отображает предписание последовательности некоторой системы операций над исходными данными с целью получения результата:
- 1) Аналитическая

- 2)Графическая
 - 3)Цифровая
 - 4)Алгоритмическая
- 27.Объект, состоящий из вершин и ребер, которые между собой находятся в некотором отношении, называют...
- 1)Системой
 - 2)Чертежом
 - 3)Структурой объекта
 - 4)Графом
- 28.Эффективность математической модели определяется ...
- 1)Оценкой точности модели
 - 2)Функцией эффективности модели
 - 3)Соотношением цены и качества
 - 4)Простотой модели
- 29.Адекватность математической модели и объекта это...
- 1)правильность отображения в модели свойств объекта в той мере, которая необходима для достижения цели моделирования
 - 2)Полнота отображения объекта моделирования
 - 3)Количество информации об объекте, получаемое в процессе моделирования
 - 4)Объективность результата моделирования
- 30.Состояние объекта определяется ...
- 1) Количеством информации, полученной в фиксированный момент времени
 - 2)Множеством свойств, характеризующим объект в фиксированный момент времени относительно заданной цели
 - 3) Только физическими данными об объекте
 - 4) Параметрами окружающей среды
- 31.Изменение состояния объекта отображается в виде...
- 1)Статической модели
 - 2)Детерминированной модели
 - 3)Динамической модели
 - 4)Стохастической модели
- 32.Фазовое пространство определяется ...
- 1)Множеством состояний объекта, в котором каждое состояние определяется точкой с координатами эквивалентными свойствам объекта в фиксированный момент времени
 - 2)Координатами свойств объекта в фиксированный момент времени
 - 3)Двумернымпространствомс координатами x, y
 - 4)Линейнымпространством
- 33.Фазовая траектория это
- 1)Вектор в полярной системе координат
 - 2)След от перемещения фазовой точки в фазовом пространстве
 - 3)Монотонно убывающая функция
 - 4)Синусоидальная кривая с равными амплитудами и частотой
- 34.Точка бифуркации это...
- 1)Точка фазовой траектории,характеризующая изменение состояния объекта
 - 2)Точка на траектории, характеризующая состояние покоя
 - 3)Точка фазовой траектории, предшествующая резкому изменению состояния объекта
 - 4)Точка равновесия
- 35.Декомпозиция это ...
- 1)Процедура разложения целого на части с целью описания объекта
 - 2)Процедура объединения частей объекта в целое
 - 3)Процедура изменения структуры объекта
 - 4)Процедура сортировки частей объекта

36. Установление равновесия между простотой модели и качеством отображения объекта называется...
- 1) Дискретизацией модели
 - 2) Алгоритмизацией модели
 - 3) Линеаризацией модели
 - 4) Идеализацией модели
37. Имитационное моделирование ...
- 1) Воспроизводит функционирование объекта в пространстве и времени
 - 2) Моделирование, в котором реализуется модель, производящая процесс функционирования системы во времени, а также имитируются элементарные явления, составляющие процесс
 - 3) Моделирование, воспроизводящее только физические процессы
 - 4) Моделирование, в котором реальные свойства объекта заменены объектами – аналогами
38. Планирование эксперимента необходимо для...
- 1) Точного предписания действий в процессе моделирования
 - 2) Выбора числа и условий проведения опытов, необходимых и достаточных для решения поставленной задачи с требуемой точностью
 - 3) Выполнения плана экспериментирования на модели
 - 4) Сокращения числа опытов
39. Модель детерминированная ...
- 1) Матрица, детерминант которой равен единице
 - 2) Объективная закономерная взаимосвязь и причинная взаимообусловленность событий. В модели не допускаются случайные события
 - 3) Модель, в которой все события, в том числе, случайные ранжированы по значимости
 - 4) Система непредвиденных, случайных событий
40. Дискретизация модели это процедура...
- 1) Отображения состояний объекта в заданные моменты времени
 - 2) Процедура, которая состоит в преобразовании непрерывной информации в дискретную
 - 3) Процедура разделения целого на части
 - 4) Приведения динамического процесса к множеству статических состояний объекта
41. Свойство, при котором модели могут быть полностью или частично использоваться при создании других моделей
- 1) Универсальностью
 - 2) Неопределенностью
 - 3) Неизвестностью
 - 4) Случайностью
42. Непрерывно-детерминированные схемы моделирования определяют...
- 1) Математическое описание системы с помощью непрерывных функций с учётом случайных факторов
 - 2) Математическое описание системы с помощью непрерывных функций без учёта случайных факторов
 - 3) Математическое описание системы с помощью функций непрерывных во времени
 - 4) Математическое описание системы с помощью дискретно-непрерывных функций
43. Погрешность математической модели связана с ...
- 1) Несоответствием физической реальности, так как абсолютная истина недостижима!!
 - 2) Неадекватностью модели

- 3) Неэкономичностью модели
4) Неэффективностью моде

Лабораторные занятия по учебной дисциплине «Математическое моделирование и статистическая обработка данных»

Лабораторная работа № 1

Работа с библиотеками элементов AnyLogic 6:
освоение методов элементов библиотеки презентации

Цель работы: закрепление навыков работы с элементами презентации для приобретения навыков визуализации процессов моделирования.

Приобретаемые навыки:

Теоретическая часть

Элементы Java

Для разработки AnyLogic необходимо познакомиться с основными конструкциями Java, подробную информацию о которых можно получить на сайте:

</anylogic/help/index.jsp?nav=/2>

Также полный перечень математических функций можно увидеть на сайте:

</javase/6/docs/api/java/lang/Math.html>

Некоторые функции

Функция	Описание
uniform_discr (min, max)	Дискретное равномерное распределение является дискретным распределением, со значениями в интервале [min, max] и равной вероятностью генерации каждого значения, находящегося в интервале.
random()	Возвращает положительное вещественное значение, лежащее в интервале [0.0, 1.0).
pow(a, b)	Возвращает значение первого аргумента, возведенное в степень второго аргумента.
xidz(a, b, x)	Пытается поделить первый аргумент на второй. Если результат - бесконечность или не является числом, то функция возвращает третий аргумент, если нет - то возвращает результат деления.
signum(d)	Возвращает знаковую функцию (сигнум) аргумента; 0, если аргумент равен нулю, 1.0, если аргумент больше, чем ноль, -1.0, если аргумент меньше, чем ноль.
min(a, b)	Возвращает наименьший из двух аргументов.
max(a, b)	Возвращает наибольший из двух аргументов.
Exponential (lambda, min)	Генерирует значение согласно экспоненциальному распределению, где lambda -параметр формы распределения, min -минимальное значение x.

Прежде чем выполнять работу, просмотрите списки функций в справочнике </anylogic/help/>, в разделе Функции AnyLogic, чтобы получить общее представление о возможностях, которые могут быть использованы в ходе работ.

Таблица 1 – Настройка свойств для проекта на рисунке 1

Название элемента	Свойство	Значение
Red (простая переменная)	Начальное значение	0
	Тип	int
Green (про-	Начальное зна-	0

стая переменная)	чение	
	Тип	int
Blue (простая переменная)	Начальное значение	0
	Тип	int
Oval (овал)	Динамические→ Цвет заливки	new Color(Red,Green,Blue)
Event (событие)	Основные→ Действие	Red = (int)uniform_discr(0, 256); Green = (int)uniform_discr(0, 256); Blue = (int)uniform_discr(0, 256); Red = Red<255 ? Red : 0; Green = Green<255 ? Green : 0; Blue = Blue<255 ? Blue : 0;

Разберем более подробно код модели:

Red = (int)uniform_discr(0, 255); – получение случайного числа в указанном диапазоне с явным преобразованием его к целому типу и присвоению полученного значения переменной с именем Red;

Red = Red<255 ? Red : 0; – если значение Red меньше 255, то заново присвоить его переменной Red, иначе присвоить 0.

Рисунок 2 - Случайное перемещение овалов в заданной плоскости

Таблица 2 – Настройка свойств для проекта на рисунке 2

Название элемента	Свойство	Значение
Oval (овал)	Количество	6
	Цвет заливки	new Color((int)(random()*255),(int)(random()*255),(int)(random()*255))
	X	X+uniform_discr(-20,20)
	Y	Y+uniform_discr(-20,20)
Event (событие)	Период	0.2
	Режим	Циклический
	Действие	Rand= uniform_discr(-20,20); X=X+Rand; if ((X>(rectangle.getX()+rectangle.getWidth())) (X<(rectangle.getX())) { X=rectangle.getX();} Rand= uniform_discr(-20,20); Y= Y + Rand; if ((Y>(rectangle.getY()+rectangle.getHeight())) (Y<(rectangle.getY())) { Y=rectangle.getY(); }

Разберем более подробно код модели: условия, описанные в коде события Event, выполняют проверку координат X,Y на нахождение их внутри области элемента rectangle, иначе устанавливать их равными координатам верхнего левого угла rectangle.

Задание 1. Вариант 1: По щелчку на любом из элементов овал установить невидимость (использовать метод элемента setVisible(false)).

Задание 1. Вариант 2: Для каждого из шести элементов овал задавать свой диапазон, в рамках которого выбрасывается случайная величина.

Некоторые способы объявления экземпляров классов

```

new Color(100,120,150) //пример создания экземпляра без выделения переменной, используется внутри методов как переменная или в ходе операций присвоения
ShapeLine line1 = new ShapeLine(); //создание экземпляра класса
                                     //с именем line1 типа ShapeLine
ShapeOval oval = new ShapeOval(); //создание экземпляра класса
                                     //с именем oval типа ShapeOval

```

Циклы и ветвления

Синтаксис ветвлений:

```

if (Условие )
    { оператор 1;}
else if (условие 2)
    { оператор 2;}
...
else if (условие N-1)
    { оператор N-1;}
else { оператор N;}

```

Синтаксис цикла ПОКА, который выполняется до тех пор, пока условие истинно:

```
While ( условие) { оператор1;... операторN; }
```

Синтаксис цикла FOR, который выполняется до тех пор, пока условие истинно:

```
For (переменная=начальное значение; условие останова; шаг)
    {операторы}
```

Например:

```

int i,y;
y = 1;
for (i=1;i<=6;i++) { y*= i; }
for( <тип элемента> <имя> : <коллекция> )
{
    Оператор1;... ОператорN; //тело цикла, выполняемое для каждого элемента
}

```

Эта форма используется для итерирования по элементам массивов и коллекций. В том случае, если Вам нужно сделать что-либо с каждым элементом коллекции, мы рекомендуем использовать эту форму цикла, потому что она легче читается и является более компактной, а также поддерживается всеми типами коллекций (в отличие от итерирования по индексу).

Приведенный ниже цикл подсчитывает количество проданных билетов в кинотеатре. Места моделируются как Java массив seats с элементами типа boolean (значение true означает, что билет на данное место продан):

```

boolean[] seats = new boolean[600]; //объявление массива
...
int nsold = 0;
for( boolean sold : seats )
    if( sold )
        nsold++;

```

Обратите внимание, что если тело цикла содержит только одну строку кода, то фигурные скобки {...} могут быть опущены. В приведенном выше коде скобки опущены как в цикле for, так и в операторе if.

Синтаксис команды присвоения значения переменной или свойству с условием:

```
Условие ? значение1 : значение2
```

Это аналог команды: if (Условие) { оператор1 } else { оператор 2 }

Размножение однотипных объектов в момент выполнения модели

Рисунок 3 – Базовый сгруппированный объект, который будет размножаться в количестве 200 штук,

где координата Y меняется лесенкой; index – указатель на элемент массива однотипных размножаемых объектов (в примере index = 1...200); % – оператор определения остатка от деления.

Например:

index%4==0 ? 100 : index%4==1 ? 120 : index%4==2 ? 140 : index%4==3 ? 120 : 100

В текущем примере если остаток от деления 0, то Y =100, иначе если остаток от деления 1, то Y =120, иначе если остаток от деления 2, то Y =140, иначе если остаток от деления 3, то Y =120, иначе Y =100.

Рисунок 4 – Результат размножения объекта с изменяемой координатой Y по описанному выше условию

Задание 2. Вариант 1: На основе примера размножить объект (можно использовать любой элемент или сгруппированную комбинацию элементов библиотеки Презентация по выбору) по квадрату.

Задание 2. Вариант 2: На основе примера размножить объект (можно использовать любой элемент или сгруппированную комбинацию элементов библиотеки Презентация по выбору) по кругу.

Задание 2. Вариант 3: На основе примера размножить объект (можно использовать любой элемент или сгруппированную комбинацию элементов библиотеки Презентация по выбору) по синусоиде.

Задание 2. Вариант 4: На основе примера размножить объект (можно использовать любой элемент или сгруппированную комбинацию элементов библиотеки Презентация по выбору) по экспоненте.

Некоторые методы элементов панели инструментов «Презентация»

Метод	Классы	Описание	Примеры
setVisible(true или false)	Все классы, которые могут быть видимы на презентации	Установка видимости или невидимости элементов	oval1.setVisible(false); - элемент oval1 сделать невидимым
isVisible()		Определение состояния видимости или невидимости элементов	If (line.isVisible()) { line.setLineWidth(4); }
setFillColor (Color)	Класс Shape, подклассы ShapeLine и пр.; элемент Линия и пр.	Установка цвета заливки фона	curve.setFillColor (new Color(0,255,0)); - элемент curve закрасить цветом Зеленый
setColor(цвет);		Установка цвета объекта, если он однороден	Line1.setColor (new Color(255,0,0));
setEndY(координата);		Установка конца объекта по оси Y	
setEndX(координата);		Установка конца объекта по оси X	
setY(координата);		Установка начальной координаты объекта по оси Y	
setX(координата);		Установка начальной координаты объекта по оси X	

setLineWidth(толщина линии);	Класс Shape, подклассы ShapeOval и пр.; элемент Овал и пр.	Установить толщину линии или внешнего контура объекта	
setRotation(угол поворота в радианах)		Установить угол поворота в радианах	
setLineColor (цвет);		Установить цвет линии или контура объекта	
setRadiusX(значение радиуса по оси X)		Установить «горизонтальный» радиус овала	
setRadiusY(значение радиуса по оси Y)		Установить «вертикальный» радиус овала	
setRadius(значение радиуса)		Установить радиус круга	
spectrumColor (целое число, значение смещения по цветовому спектру)		Возвращает цвет в соответствии со спектром, указанным целым числом и с заданным периодом (смещением по цветовому спектру)	Poliline1.spectrumColor(1,10) Даст красный цвет:  (пример при создании полилинии в проекте «Obtaining Nice Colors», в Anylogic 6 запускается через меню Справка→Начальная страница→Примеры моделей→Обучающие модели→Презентация)
getFillColor()		Получить цвет заливки	
add(название экземпляра);	Элемент Презентация	Добавить на экземпляр контейнера указанного типа указанный в скобках экземпляр другого класса (например, на презентацию добавляют линию, овал и пр.)	

Программное создание элементов презентации в процессе работы модели

Запустите проект модели «Программное рисование линий».

Разберем более подробно код модели:

```

int i,x0,y0,x1,y1;
x0=x1=y0=y1=200;
for (i=1;i<=60;i++)
{
ShapeLine line = new ShapeLine();
line.setColor( dodgerBlue );

```

```

line.setLineWidth( 4 );
x0=x1; y0=y1;
line.setX( x0 );
line.setY(y0 );
x1=(int)(x0+i*5*sin(i));
y1=(int)(y0+i*5*cos(i));
line.setEndX( x1);
line.setEndY( y1 );
presentation.add( line );
}
x0=x1=y0=y1=200; – одновременное присвоение значения нескольким переменным;
ShapeLine line = new ShapeLine(); – создание экземпляра с именем line;
line.setColor( dodgerBlue ); – установить цвет линии в dodgerBlue;
...

```

presentation.add(line); – отобразить визуально на презентации экземпляр line с назначенными свойствами.

Задание 3. Вариант 1: На основе примера размножить программно созданный объект (линия, овал, полилиния и пр.) по параболе.

Задание 3. Вариант 2: На основе примера размножить объект (линия, овал, полилиния и пр.) по кругу.

Заданием3. Вариант 3: На основе примера размножить объект (линия, овал, полилиния и пр.) по синусоиде.

Задание 3. Вариант 4: На основе примера размножить объект (линия, овал, полилиния и пр.) по экспоненте.

Программный обход элементов презентации

Запустите пример работы модели «Find All Red Circles» (в Anylogic 6 запускается через меню Справка→Начальная страница→Примеры моделей→Обучающие модели→Презентация или из папки с лабораторной работы).

Рисунок 6 – Пример готового проекта AnyLogic по программному обходу элементов презентации (код при запуске)

Разберем более подробно код модели:

```

for( int i=0; i<="" p="">
Object o = presentation.get(i);
if( o instanceof ShapeOval ) {
ShapeOval oval = (ShapeOval)o;
if( oval.getFillColor() == red )
redCircles.add( oval );
}
}

```

presentation.size() – метод size() возвращает количество экземпляров, расположенных в объекте контейнерного типа Презентация.

presentation.get(i) – метод get(идентификатор) возвращает конкретный экземпляр по уникальному идентификатору (при помещении экземпляра на презентацию, им присваивается уникальный идентификатор как указатель в массиве).

Object o = presentation.get(i); - в данной строке создается экземпляр с названием «o» на базе класса Object и ассоциируется с i-м экземпляром, расположенном на презентации.

(o instanceof ShapeOval) – instanceof определяет принадлежность классу, в данном случае проверка принадлежит ли «o» классу ShapeOval. Возвращает истину, если принадлежит, иначе возвращает ложь.

ShapeOval oval = (ShapeOval)o; – создать экземпляр oval на базе класса ShapeOval и присвоить ему данные текущего экземпляра «o» приведенного к типу ShapeOval.

redCircles.add(oval) - метод add(экземпляр) в данном случае в коллекцию с именем redCircles позволяет добавить данные текущего экземпляра oval. Коллекция redCircles была создана с типом элементов ShapeOval.

Разберем более подробно код кнопки «+»:

```
for( ShapeOval circle : redCircles ) {  
    circle.setRadiusX( circle.getRadiusX() * 1.1 );  
    circle.setRadiusY( circle.getRadiusY() * 1.1 );  
}
```

for (ShapeOval circle : redCircles) – способ создания цикла с обходом по всем элементам коллекции, в данном случае обход элементов коллекции redCircles с помещением данных текущего элемента в переменную circle.

Рисунок 8 – Пример для реализации калейдоскопа или фейерверка (код при запуске) Обратите внимание, что при создании двух коллекций Red и Blue в их свойстве Тип элементов установлено значение ShapeLine.

Рисунок 9 - Пример для реализации калейдоскопа или фейерверка (код при наступлении события)

Задание 4. Вариант 1: сделать два круга овалов вокруг существующих линий, заставить их крутиться в противоположных направлениях с разными скоростями.

Задание 4. Вариант 2: сделать круг овалов вокруг существующих линий, заставить его крутиться и круги с четным указателем на них – становиться видимыми/ не видимыми.

Задание 4. Вариант 3:

Задание 4. Вариант 4:

Инд задание:

Движение машины по горам (по синусоиде).

Размножить машины и горы (полилинии), также реализовать движение каждой машины по своим горам.

Лабораторная работа № 2

Разработка моделей на основе диаграмм состояний (стейтчарты) в AnyLogic 6

Цель работы: научиться описывать изучаемые процессы и объекты через диаграммы состояний.

Рассмотрим порядок построения модели на известном примере подготовки к экзамену.

Постановка задачи: Необходимо представить процесс подготовки к экзамену, в котором нужно выучить 17 тем. Среднее время на прочтение одной тему колеблется в диапазоне от 30 до 40 мин. Далее по памяти воспроизводится тема, если материал забывается (эмитировать этот процесс с помощью случайной величины так что, при степени воспроизведения выученного меньше чем на 20%, нужно перечитать материал полностью, если от 20 до 50 %, то прочитать половину материала, если от 50 до 80%, то прочитать треть, иначе перейти к следующей теме). Так повторяется процесс, пока не будут освоены 17 тем.

Порядок выполнения:

Создать новую модель в Anylogic с нуля.

Добавим в проект три простых переменных для хранения времени подготовки к экзамену, количества освоенных тем и временного хранения процента запомненного материала текущей темы.

Таблица 1 – Настройка свойств

Название элемента (устанавливается в свойстве <i>Имя</i> соответствующего элемента)	Свойство	Значение
---	----------	----------

Время_подготовки (простая переменная)	Отображать имя	Отметить галочкой
	Начальное значение	0
	Тип	int
Процент_запомненного (простая переменная)	Отображать имя	Отметить галочкой
	Начальное значение	0
	Тип	int
Количество_изученных_тем (простая переменная)	Отображать имя	Отметить галочкой
	Начальное значение	0
	Тип	int

Добавим элементы диаграммы состояний, соединяя их между собой, как показано на рисунке.

Таблица 2 – Настройка свойств

Название элемента (устанавливается в свойстве <i>Имя</i> соответствующего элемента)	Свойство	Значение
Начало_работы (начало диаграммы состояний)	Отображать имя	Отметить галочкой
Изучение_темы (состояние)	Отображать имя	Отметить галочкой
	Действие при выходе	Время_подготовки=Время_подготовки + uniform_discr(30, 40)
переход_к_проверке_усвоенного_материала (переход)	Отображать имя	Отметить галочкой
	Происходит	По таймауту
	По таймауту	4
переход_к_полному_повтору_темы (переход)	Отображать имя	Отметить галочкой
	Происходит	При выполнении условия
	Условие	процент_запомненного<20
ветвление1 (ветвление)	Действие	процент_запомненного=uniform_discr(10, 100)
переход1 (переход)	Отображать имя	Отметить галочкой
	Происходит	При выполнении условия

	Условие	процент_запомненного $\geq$ 20 & процент_запомненного $<$ 50
переход2 (переход)	Отображать имя	Отметить галочкой
	Происходит	При выполнении условия
	Условие	процент_запомненного $\geq$ 50 & процент_запомненного $<$ 80
переход3 (переход)	Отображать имя	Отметить галочкой
	Происходит	При выполнении условия
	По умолчанию	
Повтор_половины_темы (событие)	Отображать имя	Отметить галочкой
	Действие при выходе	Время_подготовки=Время_подготовки + uniform_discr(30, 40)/2
Повтор_трети_темы (событие)	Отображать имя	Отметить галочкой
	Действие при выходе	Время_подготовки=Время_подготовки + uniform_discr(30, 40)/3
ветвление2 (ветвление)	Действие	количество_изученных_тем=количество_изученных_тем+1
переход_к_следующей_теме (переход)	Отображать имя	Отметить галочкой
	Происходит	По таймауту
	По таймауту	2
переход_к_следующей_теме1 (переход)	Отображать имя	Отметить галочкой
	Происходит	По таймауту
	По таймауту	1.3
переход_к_следующей_теме2 (переход)	Отображать имя	Отметить галочкой

	Проис- ходит	При выполнении условия
	Условие	количество_изученных_тем<17
переход4 (переход)	Отобра- жать имя	Отметить галочкой
	Проис- ходит	При выполнении условия
	По умолча- нию	
	Дей- ствие	процент_запомненного=100
конец_темы_изучены (конечное состояние)	Отобра- жать имя	Отметить галочкой

Задание. Создайте по описанию модель. Проанализируйте действия, которые установлены у элементов Состояния и Переходы, соотнесите их с постановкой задачи.

Задание. Запустите модель на исполнение, проверьте работу. Оцените за несколько запусков модели, сколько в среднем студент будет тратить времени на подготовку экзамены из 17 тем.

Задание. Измените проект модели так, чтобы время, затрачиваемое на изучение темы было в диапазоне от 40 до 60 мин. Выполните несколько запусков нового варианта модели. Сравните общее среднее время подготовки с результатами работы исходной модели.

Задание. Измените проект модели так, чтобы на изучении было 10 тем. Выполните несколько запусков нового варианта модели. Сравните общее среднее время подготовки с результатами работы исходной модели.

Задание. Измените проект модели так, чтобы переход к изучению следующей темы после повтора половины материала происходил с повторной проверкой усвоения материала по тем же условиям, что и при первом ветвлении. Выполните несколько запусков нового варианта модели. Сравните общее среднее время подготовки с результатами работы исходной модели.

Визуализация хода подготовки в изучаемой модели

Анализируя работу модели понятно, что нет достаточной наглядности в процессе подготовки тем. Желательно видеть ход подготовки, а именно, как часто происходит возврат на повторение тем. Для этого используем элемент панели инструментов (палитры)

Статистика.

Расположим в проекте модели элемент **Временная цветовая диаграмма** и **Набор данных**.

Установим свойства согласно таблице 3. Также, **обратите внимание**, что будет изменено одно свойство окна **Main** (класса активного объекта), а также свойства **Событий** диаграммы состояний.

Таблица 3 – Настройка свойств

Название элемента (устанавливается в свойстве <i>Имя</i> соответствующего элемента)	Свойство	Значение
chart (временная цветовая диаграмма)	Трижды нажать кнопку «Добавить цветовое соответствие». В каждом из до-	

		бавленных соответствий установить пары значений:	
		Цветовое соответствие	deepSkyBlue (голубой)
		Выражение	value==Изучение темы
		Цветовое соответствие	yellow (желтый)
		Выражение	value==Повтор половины темы
		Цветовое соответствие	red (красный)
		Выражение	value==Повтор трети темы
		Основные→Временной диапазон	1000
		Основные →Обновлять автоматически Период	1
StatechartStateDS (набор данных)	Основные →Значение по оси Y	Начало_работы. getActive SimpleState()	
	Основные →Обновлять автоматически Период	1	

Таблица 4 – Настройка свойств ранее созданных элементов

Название элемента (устанавливается в свойстве <i>Имя</i> соответ- ствующего элемента)	Свойство	Значение
Main (класс активного)	Основные →Действие при запуске	chart.addDataSet(StatechartStateDS);
Изучение_темы (состояние)	Действие при входе	StatechartStateDS.add(time(), Изучение_темы)
Повтор_половины_темы (событие)	Действие при входе	StatechartStateDS.add(time(), По- втор_половины_темы);
Повтор_трети_темы (событие)	Действие при входе	StatechartStateDS.add(time(), По- втор_трети_темы);

Для облегчения понимания диаграммы необходимо отобразить какие цвета какие состояния отражают. Для этого под диаграммой в ряд расположите 3 прямоугольника и 3 метки со следующими свойствами, описанными в таблице 5, и показанными на рисунке 8.

Таблица 5 – Настройка свойств ранее созданных элементов для проекта на рисунке 8

Название элемен- та (устанавлива- ется в свойстве <i>Имя</i> соответствую- щего элемента)	Свойство	Значение
Rectangle1 (прямоугольник)	Цвет заливки	deepSkyBlue
Text1 (текст)	Текст	Изучение темы
Rectangle2	Цвет заливки	yellow

(прямоугольник)		
Text2 (текст)	Текст	Повтор половины темы
Rectangle3 (прямоугольник)5	Цвет заливки	red
Text3 (текст)	Текст	Повтор трети темы

Задание. Измените по описанию модель. Проанализируйте действия, которые установлены у элементов **Временная диаграмма** и **Набор данных**, обратите внимание на то, как связаны элементы **Диаграммы состояний** с указанными элементами.

Задание. Запустите модель на исполнение, проверьте работу. Оцените работу диаграммы. Модифицируйте значения таймаутов у элементов **Переход_к_следующей_теме**, **Переход_к_следующей_теме1** и **Переход_к_проверке_усвоенного_материала**. Проверьте, как отобразятся эти изменения на диаграмме. Подумайте, почему?

Постановка задачи 2: Создать конечный автомат, который бы включал/отключал один из 4-х сигнальных кругов с заданным значением вероятности и с заданным условием остановки работы, когда будет выполнено 5 переключений.

Таблица 6 – Настройка свойств добавляемых элементов

Название элемента (устанавливается в свойстве <i>Имя</i> соответствующего элемента)	Свойство	Значение
Oval (овал)	Имя	Oval
Oval1 (овал)	Имя	Oval1
Oval2 (овал)	Имя	Oval2
Oval3 (овал)	Имя	Oval3
Stop (простая переменная)	Тип	int
	Начальное значение	0
Button (кнопка)	Метка	Запуск
	Действие	Stop=0; statechart1.start();
statechart1 (начало диаграммы состояний)	Отображать имя	Отметить галочкой
Начало (состояние)	Отображать имя	Отметить галочкой
	Действие при входе	oval.setFillcolor(new Color(0,255,0)); oval1.setFillcolor(new Color(0,255,0)); oval2.setFillcolor(new Color(0,255,0)); oval3.setFillcolor(new Color(0,255,0)); oval.setVisible(true); oval1.setVisible(true); oval2.setVisible(true); oval3.setVisible(true);
p1	Отображать	Отметить галочкой

(переход)	имя	
	Происходит	По таймауту
	По таймауту	1
ветвление1 (ветвление)	Имя	ветвление1
п2 (переход)	Отображать имя	Отметить галочкой
	Условие	randomTrue(0.3)
П3 (переход)	Отображать имя	Отметить галочкой
	Условие	randomTrue(0.5)
П4 (переход)	Отображать имя	Отметить галочкой
	Условие	randomTrue(0.8)
П5 (переход)	Отображать имя	Отметить галочкой
	По умолчанию	
Круг1 (состояние)	Отображать имя	Отметить галочкой
	Действие при входе	oval.setFillColor(new Color(255,0,0));
Круг2 (состояние)	Отображать имя	Отметить галочкой
	Действие при входе	Oval1.setFillColor(new Color(255,0,0));
Круг3 (состояние)	Отображать имя	Отметить галочкой
	Действие при входе	Oval2.setFillColor(new Color(255,0,0));
Круг4 (состояние)	Отображать имя	Отметить галочкой
	Действие при входе	Oval3.setFillColor(new Color(255,0,0));
П6 (переход)	Отображать имя	Отметить галочкой
	Происходит	По таймауту
	По таймауту	0.1
	Действие	Stop=Stop+1
П7 (переход)	Отображать имя	Отметить галочкой
	Происходит	По таймауту
	По таймауту	0.1
	Действие	Stop=Stop+1
П8 (переход)	Отображать имя	Отметить галочкой
	Происходит	По таймауту
	По таймауту	0.1
	Действие	Stop=Stop+1
П9 (переход)	Отображать имя	Отметить галочкой

	Происходит	По таймауту
	По таймауту	0.1
	Действие	Stop=Stop+1
П10 (переход)	Отображать имя	Отметить галочкой
	По умолчанию	
ветвление2 (ветвление)	Имя	ветвление2
П11 (переход)	Отображать имя	Отметить галочкой
	Условие	Stop==5
Конец (конечное состояние)	Отображать имя	Отметить галочкой

Задание: Модифицировать пример:

1. добавить гистограмму с четырьмя столбцами, каждый из которых будет накапливать статистику по исходам работы диаграммы (например, первый столбец будет показывать сколько раз работа закончилась окрашиванием в красный 1-го круга).
2. Запустить модель 100 раз, проанализировать результат статистики.
3. Проварьировать параметры переходов (в свойствах параметр Условие) к блокам Круг1...Круг4 так, чтобы добиться равномерного распределения статистики.

Приложение А

Порядок выполнения действий элементов диаграммы состояний

Очень важно точно знать, в каком именно порядке выполняются действия элементов диаграммы состояний. Для этого мы предлагаем Вам изучить приведенный ниже алгоритм.

При срабатывании перехода выполняются следующие действия (в указанном порядке):

1. Действия при выходе из состояния, начиная с текущего простого состояния, и дальше вверх по иерархии состояний, заканчивая тем сложным состоянием, на уровне иерархии которого и передается управление.
2. Действие перехода.
3. Действия при входе в состояние, начиная со сложного состояния, которое получает управление, и дальше, вниз по иерархии состояний, вплоть до простого состояния или псевдосостояния, в которое передается управление.
4. Если управление передается в псевдосостояние, то выполняется код действия псевдосостояния, а затем управление немедленно передается другому состоянию, и описанный выше алгоритм выполняется сначала.

Действия состояний и переходов выполняются за нулевое модельное время. Поэтому они не могут содержать синхронизационных операций и операций задержки и не могут вызывать методы, явно или неявно содержащие такие операции.

Пример

Предположим, что состояние N является текущим простым состоянием, и срабатывает переход T1. Тогда действия выполняются в следующем порядке:

1. Действие при выходе из состояния N
2. Действие при выходе из состояния M
3. Действие перехода T1
4. Действие состояния ветвления

Затем, в зависимости от дополнительных условий переходов, будет выбран переход T2 или T3. Если будет выбран переход T2, то выполняются следующие действия:

5. Действие перехода T2

6. Действие указателя начального состояния I1 (действия при входе и выходе из состояния L не выполняются, поскольку управление остается в этом состоянии)
7. Действие при входе в состояние M
8. Действие указателя начального состояния I2
9. Действие при входе в состояние N

Если выбирается переход T3, то выполняются следующие действия:

10. Действие при выходе из состояния L
11. Действие при выходе из состояния K (действия состояния V не выполняются)
12. Действие перехода T3
13. Действие при входе в состояние S
14. Действие при входе в состояние P
15. Действие состояния-ветвления
16. Действие при выходе из состояния P
17. Действие перехода T4 (дополнительное условие перехода должно быть равно true, поскольку это единственный выход из ветвления)
18. Действие при входе в состояние Q
19. Действие указателя начального состояния I3
20. Действие при входе в состояние R

Лабораторная работа № 3

Разработка моделей динамических систем в AnyLogic 6

Цель работы: научиться описывать изучаемые процессы динамических систем.

Приобретаемые навыки:

Модель системы с управлением

Познакомьтесь с работой модели, описанной в книге [Карпова Ю. имитационное моделирование систем. Введение в моделирование с AnyLogic 5] и расположенной в папке с названием файла *ControlSystem.alp*.

Рассмотрим в качестве примера классическую динамическую систему, состоящую из двух связанных подсистем: объекта управления и регулятора.

Рисунок – Структура системы регулирования

Пусть *объектом управления* является бойлер, который нагревается до температуры T . Величину температуры T необходимо поддерживать на заданном уровне T_z . Температура бойлера зависит от входного параметра — в нашем случае от мощности тока U , подаваемого на нагревательный элемент бойлера. Объект управления подвергается внешнему возмущению f (f может характеризовать внешнюю температуру или теплоизоляцию), вследствие чего значение выходного параметра T может меняться. Поддержание значения T на заданном уровне T_z есть задача регулятора.

Регулятор по разнице заданного и текущего значений выходного параметра ($T_z - T$) формирует величину входного параметра объекта управления — в нашем случае регулируется мощность U , подаваемая на нагревательный элемент.

Уравнения функционирования как бойлера, так и регулятора будем записывать явно, а не с помощью передаточных функций.

Объект управления. Выходная величина T типичного объекта управления описывается уравнением:

$$dT/dt = 1/a * (k*U - F - T),$$

где:

a — коэффициент пропорциональности, определяемый параметрами объекта управления (выберем его значение 10);

k — коэффициент усиления объекта по управлению (установим его в 1);

F — изменяющееся внешнее воздействие (температура окружающей среды или теплоизоляция).

Регулятор можно строить разными способами; мы будем использовать здесь так называемый регулятор с пропорционально-интегральным законом управления (ПИ-регулятор). Выходную величину (управление) регулятора определим так:

$$U = U_i + K_p \cdot (T_z - T),$$

где:

U — управление, которое состоит из интегральной и пропорциональной части;

U_i — интегральная составляющая управления;

K_p — коэффициент при пропорциональной составляющей управления ($=1$).

Интегральная составляющая управления может быть задана так:

$$dU_i/dt = K_i \cdot (T_z - T),$$

где:

K_i — коэффициент при интегральной составляющей управления ($=0.3$).

Поставим задачу исследовать с помощью модели качество регулирования температуры бойлера в зависимости от соотношений параметров объекта управления и регулятора.

Для моделирования системы управления можно в поле редактора определить все переменные и параметры этой системы, и для каждой переменной в соответствующем поле окна свойств этой переменной записать правую часть определяющего ее уравнения. Для нашей системы управления, описываемой соотношениями:

$$d(T)/dt = 1/a \cdot (k \cdot U - F - T),$$

$$U = U_i + K_p \cdot (T_z - T),$$

$$d(U_i)/dt = K_i \cdot (T_z - T),$$

такая модель с названием *ControlSystem.alp* представлена в папке.

Переменные T и U_i определяются как **накопители**, правая часть дифференциальных уравнений записывается в соответствующем поле в виде выражения.

В модели *ControlSystem* используются два режима для задания различного поведения внешней среды, определяемой значением параметра f . Переключение режимов в моделях динамических систем легко осуществляется с помощью стейтчарта. В нашей модели стейтчарт с именем *statechart* имеет два состояния. В одном состоянии никаких соотношений для F не определено, и, следовательно, эта переменная определена своим постоянным (начальным или текущим) значением. В другом состоянии F определено уравнением $F = F_{\max} \cdot (1 + \sin(F_w \cdot t)) / 2$. Переход в это состояние осуществляется, как только логическая переменная $F_{\sin}$ станет истинной (в результате внешнего воздействия).

Использование стейтчартов для переключения моделей позволяет легко строить сложные гибридные модели динамических систем, т. е. такие модели, в которых непрерывное поведение прерывается дискретными событиями, вызывающими изменение поведения.

С моделью *ControlSystem* можно выполнять различные эксперименты, меняя ее параметры в окне дерева объектов модели.

Анимация поведения системы в AnyLogic позволяет более ясно представить процессы, происходящие в системе в их совокупности.

В окне анимации динамика системы отражена изменением цвета графических объектов, их позицией и размерами. С помощью флажка на анимации с именем **Колебания** можно переключиться на режим изменения внешнего воздействия F по синусоидальному закону.

Задание: В режиме проекта модели просмотрите все объекты модели и их свойства, узнайте и объясните связь свойств анимации с сутью модели.

Задание: Запустите модель. Проанализируйте поведение, изменяя значения параметров модели с помощью бегунков, связанных с параметрами модели.

Модель странного аттрактора динамических систем

Рассмотрим пример построения модели для диссипативных систем из работы [Анищенко В.С. Динамические системы. — db/msg.html?mid=1186976&uri=text3.html].

Постановка задачи:

Движения диссипативных (Иными словами, это устойчивое состояние, возникающее в неравновесной среде при условии диссипации (рассеивания) энергии, которая поступает извне. Диссипативная система иногда называется ещё *стационарной открытой системой* или *неравновесной открытой системой*) систем целесообразно разделить на два класса: класс переходных, нестационарных движений, отвечающих релаксации от начального к предельному множеству состояний, и класс установившихся стационарных движений, фазовые траектории которых целиком принадлежат предельным множествам. Важными с физической точки зрения являются притягивающие предельные множества — *аттракторы*. С течением времени произвольное начальное состояние из некоторой области притяжения G , включающей в себя аттрактор G_0 , релаксирует к G_0 . Движение, которому отвечает фазовая траектория в области притяжения, есть переходной процесс. Установившееся движение характеризуется принадлежностью фазовых траекторий предельному множеству, то есть аттрактору G_0 .

Важным результатом исследований последних лет явилось обнаружение принципиально новых типов движений в динамических системах. Таким движениям в фазовом пространстве размерности соответствуют сложным образом устроенные притягивающие множества, траектории изображающих точек которых не принадлежат ни к одному из описанных выше типов аттракторов. Фазовые траектории представляются здесь в виде бесконечной, нигде не пересекающейся линии.

Впервые подобные свойства динамической системы в 1963 году обнаружил Э. Лоренц при численном исследовании динамики трехмерной модели тепловой конвекции. Спустя восемь лет в теоретической работе Д. Рюэля и Ф. Такенса притягивающая область в фазовом пространстве динамической системы, характеризуемая режимом установившихся непериодических колебаний, была названа *странным аттрактором*. Этот термин был сразу воспринят исследователями и утвердился для обозначения математического образа режима нерегулярных колебаний детерминированных динамических систем.

Примером системы с хаотическим аттрактором являются уравнения генератора с инерционной нелинейностью (генератора Анищенко-Астахова). Эта система является обобщением уравнений Ван дер Поля на случай трехмерного пространства:

Результаты численного решения уравнения для значений параметров $m = 1,5$, $g = 0,2$ приведены на рис. выше, который также иллюстрирует хаотический аттрактор.

Порядок выполнения:

Создать новую модель в Anylogic с нуля.

Добавим в проект с палитры **Системная динамика** три **Накопителя** для координат модели меняющихся во времени, две **Динамических переменных** как параметров модели (можно использовать просто элементы **Параметр**, так как во время процесса работы модели их значения меняться не будут). С палитры **Основная** добавить одну **Функцию**, которая будет принимать значение в зависимости от условий. Также для просмотра результатов моделирования в динамике с палитры **Статистика** добавить три элемента **График**.

Соединители между накопителями и переменными появятся автоматически при задании уравнений накопителей.

Таблица 1 – Настройка свойств для проекта

Название элемента (устанавливается в свойстве <i>Имя</i> соответствующего элемента)	Свойство	Значение
m (динамическая пе-	Отображать имя	Отметить галочкой
	m=	1.5

ременная)				
g (динамическая переменная)	Отображать имя	Отметить галочкой		
	g=	0.2		
x (накопитель)	Отображать имя	Отметить галочкой		
	Начальное значение	1		
	d(x)/dt=	m*x+y-x*z		
y (накопитель)	Отображать имя	Отметить галочкой		
	Начальное значение	0		
	d(y)/dt=	-x		
I (функция)	Тип возвращаемого значения	int		
	Аргументы функции:	x	double	
	Тело функции	if (x<=0) return 0; else return 1;		
z (накопитель)	Отображать имя	Отметить галочкой		
	Начальное значение	0		
	d(z)/dt=	-g*z+g*I(x)*x*x		
Plot (график)	Горизонтальная шкала	Фиксированный	-15	15
	Вертикальная шкала	Фиксированный	-15	15
	Выполнить действие!!! нажать на кнопку «Добавить элемент данных», в появившихся свойствах установить следующие значения:			
	Значение по оси x	x		
	Значение по оси y	y		
	Цвет	violetRed		
	Заголовок	ось абсцисс = x; ось ординат =y;		
Plot1 (график)	Горизонтальная шкала	Фиксированный	-15	15
	Вертикальная шкала	Фиксированный	-15	15
	Выполнить действие!!! нажать на кнопку «Добавить элемент данных», в появившихся свойствах установить следующие значения:			
	Значение по оси x	x		
	Значение по оси y	z		
	Цвет	lime		
	Заголовок	ось абсцисс = x; ось ординат =z;		

Plot2 (график)	Горизонтальная шкала	Фиксированный	-15	15
	Вертикальная шкала	Фиксированный	-15	15
	Выполнить действие!!! нажать на кнопку «Добавить элемент данных», в появившихся свойствах установить следующие значения:			
	Значение по оси x	z		
	Значение по оси y	y		
	Цвет	blue		
	Заголовок	ось абсцисс = z; ось ординат =y;		

Задание. Создайте по описанию модель. Проанализируйте систему уравнений, и как теоретически должны влиять параметры x, y, z друг на друга, а также мера влияния параметров g, m на x, y, z .

Задание. Запустите модель на исполнение, проверьте работу. Оцените за какой промежуток времени система перейдет в стационарное состояния (изменения траекторий будут минимальны).

Задание. Создайте план выполнения модели, определите диапазоны изменений параметров g, m так, чтобы координаты x, y, z не выходили за границы координат $[-15, 15]$. Составьте таблицу проведения экспериментов, в которой будут отмечены значения g, m и время t , при котором процесс становится стационарным.

Задание. Проведите эксперимент с отрицательными значениями g, m . Какие отличия в результате моделирования наблюдаются.

Самостоятельное задание 1. Создать модель гиперболической спирали и отразить процесс ее рисования на графике с координатными осями x, y в режиме автоматических горизонтальной и вертикальной шкал. Провести несколько экспериментов с разными значениями параметра: **a**. Сделать отчет с полученными графиками.

$$x = a * \cos(t)/t,$$

$$y = a * \sin(t)/t$$

Подсказка: для организации передачи времени в уравнения сделайте **Накопитель**, например с именем t_* , у которой в свойстве $d(t_*)/dt$ установить **1**. А x и y сделать **динамическими переменными**, в которых вместо t поставить t_* .

Самостоятельное задание 2. Создать модель **Архимедовой спирали** и отразить процесс ее рисования на графике с координатными осями x, y в режиме автоматических горизонтальной и вертикальной шкал. Провести несколько экспериментов с разными значениями параметра: **a**. Сделать отчет с полученными графиками.

В полярной системе координат спираль:

где

a - шаг спирали, ϕ - угол в радианах.

Для перехода к декартовым координатам используются формулы:

Подсказка: для организации динамики в качестве значения угла ϕ сделайте **Накопитель**, например с именем f , у которой в свойстве $d(f)/dt$ установить **1**. А x и y сделать **динамическими переменными**, в которых вместо ϕ поставить f .

Самостоятельное задание 3. Создать модель **Трактрисы (линии влечения)** — плоская трансцендентная кривая, для которой длина отрезка касательной от точки касания до точки пересечения с фиксированной прямой является постоянной величиной. Такую линию описывает предмет, волоочащийся на верёвке длины a за точкой, движущейся по оси абсцисс. Трактриса также является кривой погони.

Отразить процесс ее рисования на графике с координатными осями x,y в режиме автоматических горизонтальной и вертикальной шкал. Провести несколько экспериментов с разными значениями параметра: **a**. Сделать отчет с полученными графиками.

Подсказка: для организации передачи времени в уравнения сделайте **Накопитель**, например с именем **t_**, у которой в свойстве **d(t_)/dt=установить 1**. А **x** и **y** сделать **динамическими переменными**, в которых вместо **t** поставить **t_**.

Лабораторная работа № 4

Сети массового обслуживания в среде AnyLogic

Цель работы – изучение средств и возможностей программы AnyLogic для построения модели сети массового обслуживания.

Таблица 1 – Основные элементы для создания модели

Элемент	Назначение	Настройки по умолчанию	Методы элемента	Свойства элемента	
Source	Генерация заявок на вход СМО	Entity	void inject(int n).	Создает n заявок.	intarrivalType - Способ генерации заявок
			int count().	Возвращает количество заявок, созданных объектом	doublerate- Интенсивность прибытия заявок
					EntitynewEntity- Новая заявка
					voidonExit – Действие при выходе
Queue	Моделирование очереди заявок		intsize()	Возвращает количество заявок, находящихся в данный момент в очереди	intcapacity – Емкость очереди
					booleanenableTimeout – Разрешить уход по таймауту
					doubletimeout – Таймаут
			booleancanEnter()	Возвращает true, если в очередь может быть добавлена еще одна заявка.	voidonExitTimeout – Действие при уходе по таймауту
					booleanenablePreemption - Разрешить вытеснение
					doublepriority – Приоритет заявки
					voidonExitPreempted – Действие при вытеснении
					voidonEnter – Действие при входе
					voidonExit – Действие при выходе
Delay	Моделирование процессора обслуживания		intsize()	Возвращает количество заявок, задерживае-	intcapacity – Вместимость объекта Delay. Задает максимальное количество

				мых объектом в текущий момент времени.	заявок, которое может одновременно находиться в объекте. Значение по умолчанию равно одной заявке.
					voidonEnter – Действие при входе
			booleancanEnter()	Возвращает true, если объект может принять еще одну заявку.	voidonExit – Действие при выходе
					doubledelayTime – Время задержки заявки
Service	Моделирование очереди с процессором обслуживания с учетом использованных ресурсов.		intqueueSize()	Возвращает количество заявок во вложенном объекте Queue.	intquantity – Выражение, вычисляющее количество ресурсов, необходимое текущей заявке.(по ум. равно 1)
					voidonEnterDelay – Действие при начале задержки
			intdelaySize()	Возвращает количество заявок во вложенном объекте Delay.	ResourcePoolresourcePool – Выражение, возвращающее имя объекта ResourcePool, задающего ресурсы, которые требуются заявкеЗначение по умолчанию null.
					intquantity – Количество ресурсов, захваченных заявкой.
					doubledelayTime – Время задержки, вычисленное для заявки
Sink	Уничтожение заявок, после их выхода из канала СМО		intcount()	возвращает количество заявок, уничтоженных объектом Sink.	voidonEnter – Действие при входе. Код, выполняемый при поступлении заявки в объект.
SelectOutput	Используется для выбора канала СМО при моделировании.				booleanconditionIsProbabilistic – Определяет, как будет производиться маршрутизация заявок
					booleancondition – Условие
					codeonEnter –

					Действие при входе
					codeonExitTrue – Действие при выходе true
					codeonExitFalse – Действие при выходе false

В качестве примера разберем пример банковского обслуживания клиентов.

Задание: В банковском отделении находятся банкомат и стойки банковских кассиров, которые предназначены для быстрого и эффективного обслуживания посетителей банка. Операции с наличностью клиенты банка производят с помощью банкомата, а более сложные операции, такие как оплата счетов, – с помощью кассиров. Необходимо произвести оценку затрат операций и определить, сколько денег тратится на обслуживание одного клиента и какую часть этой суммы составляют накладные расходы на оплату работы персонала банка, а какую – на обслуживание посетителей.

Порядок выполнения:

Создать новую модель в Anylogic с нуля.

Рисунок 3 – Модель банкомата

Добавим в проект четыре объекта: source, queue, delay, sink. Подробное описание объектов приводится ниже. Соединим каждый объект друг с другом как показано на рисунке выше, с помощью соединителя. Объект находится во вкладке «Основные».

Объект source генерирует заявки (entities) определенного типа через заданный временной интервал. Заявки представляют собой объекты, которые производятся, обрабатываются, обслуживаются или еще каким-нибудь образом подвергаются воздействию моделируемого процесса: это могут быть клиенты в системе обслуживания, детали в модели производства, документы в модели документооборота и др. В нашем примере заявками будут посетители банка, а объект source будет моделировать их приход в банковское отделение.

Объект queue моделирует очередь клиентов, ожидающих обслуживания. Объект delay моделирует задержку. В нашем примере он будет имитировать банкомат, тратящий определенное время на обслуживание клиента. Объект sink обозначает конец блок-схемы.

Для каждой модели, созданной в EnterpriseLibrary, автоматически создается блок-схема с наглядной визуализацией процесса, с помощью которой вы можете изучить текущее состояние модели, например, длину очереди, количество обслуженных человек и т.д.

Теперь зададим данные модели, изменив свойства созданных объектов. В свойстве interarrivalTime объекта source укажите – **exponential(0.67)**.

Рисунок 4 – Настройка элемента source

Данное значение будет отображать, как часто клиенты заходят в банк. Функция **exponential()** является стандартной функцией генератора случайных чисел AnyLogic.

В свойстве capacity объекта queue, задайте максимальную длину очереди – 15.

Рисунок 5 – Настройка элемента queue

В свойстве delayTime объекта delay задайте время задержки (время обслуживания) – **triangular(0.8, 1, 1.3)**. Значение 0.8 – минимальное для обслуживания клиента, 1 – среднее, 1.3 – наибольшее время для обслуживания.

Объект sink настраивать не нужно.

Запустим модель. Для запуска модели необходимо нажать F5 или нажать в панели инструментов зеленую стрелку.

Усложним модель, добавив в нее банковских кассиров. Можно моделировать число кассиров, как и банкомат, с помощью объектов delay. Но куда более удобным представляется моделирование числа кассиров с помощью ресурсов. Ресурс – это специальный объект EnterpriseLibrary, который может потребоваться заявке для выполнения какой-то

задачи. В нашем примере посетителям банковского отделения (заявкам) необходимо получить помощь у банковских служащих (ресурсов).

Добавьте на диаграмму следующие объекты: `selectOutput`, `Service`, `ResourcePool`, подробное описание элементов и их настройка приводится ниже. Соедините элементы с помощью объекта «Соединитель», так как показано на рисунке 8.

`SelectOutput` – является блоком принятия решения. В зависимости от заданного вами условия, заявка, поступившая в этот объект, будет поступать на один из двух выходов объекта. Оставьте свойство **`selectCondition – uniform() < 0.5`**, тогда к кассирам и банкомату будет приходить примерно равное количество клиентов;

`Service` – моделирует занятие заявкой ресурса на определенное время. С помощью этого объекта мы промоделируем обслуживание клиента кассиром. Задайте следующие свойства объекта: назовите объект `tellerLines`(свойства, имя).

Далее укажите, что в очереди к кассирам может находиться до 20 человек (свойство `queueCapacity`);

Задайте время обслуживания (свойство `delayTime`). Будем полагать, что время обслуживания имеет треугольное распределение с минимальным средним значением 2.5, средним – 6 и максимальным – 11 минут;

`ResourcePool` – задает ресурсы определенного типа. Он должен быть подсоединен к объектам, моделирующим занятие и освобождение ресурсов (в нашем случае это объект `Service`).

Задайте следующие свойства объекта: назовите объект `tellers`;

Задайте число кассиров (свойство `capacity`) – 4.

Измените имя объекта `delay` на `АТМ` (банкомат).

Модель должна выглядеть следующим образом:

Запустите модель и изучите ее поведение.

Добавим в модель возможность сбора и просмотра статистики. Создадим класс сообщения **`Customer`**. Сообщения этого класса будут представлять клиентов банковского отделения. Выберите базовый класс `Entity` (сообщения), добавьте параметры для хранения информации о проведенном времени:

1) в панели Проект, щелкните правой кнопкой мыши по элементу модели и выберите Создать | Java класс из контекстного меню.

Сделаем так, чтобы этот класс наследовался от базового класса заявки `Entity`, выберем из выпадающего списка Базовый класс полное имя данного класса:

`com.xj.anylogic.libraries.enterprise.Entity`;

На второй странице зададим следующие параметры:

`enteredSystem` типа **`double`** для сохранения момента времени, когда клиент пришел в банковское отделение;

- **`startWaiting`** типа **`double`** для сохранения момента времени, когда клиент встал в очередь к банкомату;

Рисунок 18 - Настройка Java класса

Теперь добавим палитры «Статистика» элемент «Данные гистограммы». Настроим его следующим образом:

- Изменим Имя на `waitTimeDistr`.

- Сделаем количество интервалов равным 50.

- Зададим начальный размер интервала: 0.01;

Создайте еще один такой же элемент с теми же свойствами. Измените Имя этого элемента на `timeInSystemDistr`.

Изменим свойства некоторых элементов.

Блок `source`

- Свойство Новая заявка – введите **`newCustomer()`**.

- Введите **`Customer`** в поле Класс заявки.

- Введите **entity.enteredSystem = time()**; в поле Действие при выходе. Функция **time()** возвращает текущее значение модельного времени;
Блок **tellerLines** (блок **Service**)

- Введите **Customer** в поле Класс заявки.

- Действие при входе: **entity.startWaiting = time()**;

- Действие при выходе: **waitTimeDistr.add(time() - entity.startWaiting)**;

Блок **queue**

- Введите **Customer** в поле Класс заявки

- Действие при входе: **entity.startWaiting = time()**;

- Действие при выходе: **waitTimeDistr.add(time() - entity.startWaiting)**;

Рисунок 22 - Настройка элемента **queue**

Блок **АТМ** (блок **delay**)

- Введите **Customer** в поле **Класс заявки**;

Блок **sink**

- Введите **Customer** в поле Класс заявки.

- Действие при входе: **timeInSystemDistr.add(time() - entity.enteredSystem)**; Данный код добавляет полное время пребывания клиента в банковском отделении в объект сбора данных гистограммы **timeInSystemDistr**.

Рисунок 24 - Настройка элемента **sink**

Добавьте две гистограммы для отображения распределений времен ожидания клиента и пребывания клиента в системе. Чтобы добавить гистограмму на диаграмму класса активного объекта, перетащите элемент Гистограмма из палитры Статистика в то место графического редактора, куда вы хотите ее поместить.

Для добавления данных в гистограмму нажмите кнопку «Добавить данные». В поле «Данные» введите имя соответствующего элемента – **waitTimeDistr.** (Рисунок 25)

Во второй диаграмме, в поле «Данные» введите – **timeInSystemDistr.**

Запустите модель.

Практическое задание

Сервер обрабатывает запросы, поступающие с автоматизированных рабочих мест с интервалами, распределенными по показательному закону со средним значением 2 мин. Время обработки сервером одного запроса распределено по экспоненциальному закону со средним значением 3 мин. Сервер имеет входной буфер емкостью 5 запросов.

1. Построить имитационную модель для определения математического ожидания времени и вероятности обработки запросов.

Порядок выполнения.

Создадим модель с нуля, следующим образом.

Измените свойства объекта **source** так что бы время между прибытиями заявок было 120 секунд. (**exponential 1/120.0**).

Измените свойства объекта **queue**:

- Задайте длину очереди. Введите в поле Вместимость 5. В очереди будут находиться не более 5 запросов.
- Установите флажок Включить сбор статистики, чтобы включить сбор статистики для этого объекта. В этом случае по ходу моделирования будет собираться статистика по количеству запросов в очереди. Если же вы не установите этот флажок, то данная функциональность будет недоступна, поскольку по умолчанию она отключена для повышения скорости выполнения модели.

Измените свойства объекта **delay**.

- Обработка одного запроса занимает примерно 3 мин. Задайте время обслуживания, распределенное по экспоненциальному закону со средним значением 3 мин. Для этого введите в поле Время задержки **exponential(1/180.0)**. Функция **exponential()** является стандартной функцией генератора случайных чисел **AnyLogic**. **AnyLogic**

предоставляет функции и других случайных распределений, таких как нормальное, треугольное, и т. д.

- Установите флажок Включить сбор статистики.

2. Создайте анимацию работы сети.

Нарисуйте прямоугольник, который будет обозначать на анимации сервер. Для этого необходимо зайти во вкладку Презентация и выбрать элемент прямоугольник.

Измените динамические свойства прямоугольника так, чтобы он менял цвет на красный в зависимости от того, обрабатывает ли сервер в данный момент времени запрос и на зеленый если нет.

Например для того, чтобы прямоугольник менял цвет на красный необходимо дополнить свойства следующим кодом:

delay.size()>0?red:white

Нарисуйте линию, которая будет отображать очередь запросов на обработку.

Чтобы нарисовать ломаную, сделайте двойной щелчок мышью по элементу Ломаная в палитре во вкладке Презентация.

Задайте ломаную линию в качестве фигуры анимации очереди. На странице свойств объекта queue, введите **polyline** в поле Фигура анимации.

Задайте прямоугольник в качестве фигуры анимации сервера(delay). Введите в поле Фигура анимации имя нашего прямоугольника: **rectangle**.

Выберите из выпадающего списка Тип анимации Одиночная.

Постройте диаграмму для отображения статистики занятости сервера и длины очереди.

Перетащите элемент Столбиковая диаграмма из палитры Статистика на диаграмму класса.

- Измените Заголовок на **SERVER utilization**.
 - Введите **delay.statsUtilization.mean()** в поле Значение. Здесь delay - это имя нашего объекта delay. У каждого объекта delay есть встроенный набор данных statsUtilization, занимающийся сбором статистики использования этого объекта. Функция **mean()** возвращает среднее из всех измеренных этим набором данных значений. Вы можете использовать и другие методы сбора статистики, такие, как **min()** или **max()**.
 - Измените внешний вид диаграммы.
Аналогично создайте диаграмму для отображения длины очереди.
 - Измените Заголовок на **Queue length**.
 - Введите **queue.statsSize.mean()** в поле Значение.
 - Измените внешний вид диаграммы.
- Проверьте работоспособность модели.

Лабораторная работа №5

Клеточные автоматы

Цель работы: Изучение средств и возможностей программы AnyLogic для построения модели клеточного автомата.

Практическое задание

Задание: Построить модель распространения вирусов в компьютерной системе. Системные файлы представим в виде квадратов.

Квадрат, окрашенный в зелёный цвет – не зараженный файл.

В красный - зараженный.

Синий – вылечен антивирусом.

Создайте модель с нуля.

Добавьте из основной палитры элемент «популяция агентов».

Нажимаем далее и выбираем упорядоченное расположение элементов и нажимаем готово.

Настроим Main, класс активного объекта.

Добавим следующее действие при запуске:

```
for(int i=0; i<10; i++) //запускаем цикл для отправки сообщения о заражении файлов.  
cell.get( uniform_discr( 999 ) ).receive( "Infection" );//самосообщение
```

Добавим в класс активного объекта(Virus) следующую схему:

Для создания схемы используйте палитру «Диаграмма состояний»

- Элемент statechart – начало диаграммы состояния.
- Элементы Not_infected, Infected, Cured – состояние.
- Элементы Infection1, Contact1, EndOfIllness1, EndOfImmunity1 – переходы.

Элемент Not_infected:

- Действие при входе: **rectangle.setFill(chartreuse);**

Элемент Infected:

- Действие при входе: **rectangle.setFill(orangeRed);**

Элемент Cured:

- Действие при входе: **rectangle.setFill(cornflowerBlue);**

Строка **rectangle.setFill(chartreuse);** посылает нашему элементу (файлу), какой цвет ему принять, в данном случае ярко-зеленый (chartreuse), так как файл еще не заражен.

Строка **rectangle.setFill(orangeRed);** посылает нашему элементу (файлу), какой цвет ему принять, в данном случае красный, так как файл поражен вирусом.

Строка **rectangle.setFill(cornflowerBlue);** посылает нашему элементу (файлу), какой цвет ему принять, в данном случае синий, так как файл вылечен антивирусом.

Переход Infection1:

Переход осуществляется при получении сообщения, которое ранее описывалось в классе Main.

Переход Contact1:

Переход осуществляется с определенной интенсивностью.

Строка **get_Main().InfectionProbability** получает из поля класса Main, значение из переменной InfectionProbability.

Строка **send("Infection", RANDOM_CONNECTED);** отправляет сообщение о заражении файлов случайно.

Переход EndOfIllness1:

Строка **uniform(get_Main().InfectiousMin, get_Main().InfectiousMax)** равномерно получает данные из переменной InfectiousMin и InfectiousMax.

Переход EndOfImmunity1:

Строка **uniform(get_Main().ImmuneMin, get_Main().ImmuneMax)** равномерно получает данные из переменных ImmuneMin и ImmuneMax.

Добавим несколько элементов для управления временем, количеством незараженных и зараженных файлов, вероятностью заражения вирусом файлов.

Используйте переменные и бегунок из палитры «Элементы управления»:

Переменная LatencyMin используется для определения наименьшего времени задержки распространения вируса.

Переменная LatencyMax используется для определения наибольшего времени задержки распространения вируса.

Переменная InfectiousMin используется для определения наименьшего числа зараженных файлов.

Переменная InfectiousMax используется для определения наибольшего числа зараженных файлов.

Переменная ImmuneMin используется для определения наименьшего числа незараженных файлов.

Переменная `ImmuneMax` используется для определения наибольшего числа не зараженных файлов.

Переменная `InfectionProbability` используется для определения вероятности заражения.

Настроим их следующим образом:

Настройка переменных.

Переменная `LatencyMin`:

- Тип – `double`
- Значение по умолчанию – 3

Переменная `LatencyMax`:

- Тип – `double`
- Значение по умолчанию – 6

Переменная `InfectiousMin`:

- Тип – `double`
- Значение по умолчанию – 7

Переменная `InfectiousMax`:

- Тип – `double`
- Значение по умолчанию – 15

Переменная `ImmuneMin`:

- Тип – `double`
- Значение по умолчанию – 30

Переменная `ImmuneMax`:

- Тип – `double`
- Значение по умолчанию – 70

Переменная `InfectionProbability`:

- Тип – `double`
- Значение по умолчанию – 2

Настроим агент `cell` для сбора статистики следующим образом.

Строка `item.statechart.isStateActive( Virus.Not_infected )` берет данные из активного класса `Virus`, из состояния `Not_infected`.

Строка `item.statechart.isStateActive( Virus.Infected )` берет данные из активного класса `Virus`, из состояния `Infected`.

Строка `item.statechart.isStateActive( Virus.Cured )` берет данные из активного класса `Virus`, из состояния `Cured`.

Добавим временную диаграмму с накоплением из палитры «Статистика» и настроим ее

Строка `cell.NInfectious()`, означает что диаграмма берет данные из функции `NInfectious()` сбора статистики описанной ранее в агенте `cell`.

Строка `cell.NSusceptible()`, означает что диаграмма берет данные из функции `NSusceptible()` сбора статистики описанной ранее в агенте `cell`.

Строка `cell.NImmune()`, означает что диаграмма берет данные из функции `NImmune()` сбора статистики описанной ранее в агенте `cell`.

Запустите модель.

Лабораторная работа №6

Многоагентный подход в среде AnyLogic.

Цель работы: Изучение средств и возможностей программы AnyLogic для построения многоагентных моделей.

Практическое задание

Задание 1: Создайте демонстрацию стандартных типов сетей агентов и типов их расположения (в непрерывном пространстве), поддерживаемых AnyLogic. Типы сетей включают в себя случайную сеть, сеть, формируемую согласно расстоянию между аген-

тами, малый мир, безразмерную сеть и решеточно упорядоченное кольцо. Типы расположения - случайное, упорядоченное, кольцо и пружинящая масса. Настройте модель так, что бы было возможно динамически менять количество моделируемых агентов, а также их расположение и сеть.

Создайте модель с нуля.

Добавьте элемент «Популяция агентов» с палитры «Основные».

Настройте следующим образом.

Нажмите далее и задайте ширину и высоту – 500.

Далее создадим линию для соединения сети и компьютеры.

Перейдем к классу Network и удалим созданный прямоугольник.

Перенесем из палитры «Презентация» овал:

- Радиус по X и Y – 2
- Цвет заливки – черный
- Цвет линии - черный

Перенесем к нему линию (Рисунок 7).

Настроим объект:

- Вкладка «Динамические», количество - **getConnectionsNumber()**
- Параметр dX - **getConnectedAgent(index).getX() - getX()** – означает что мы размещаем точки каждой строки в точках подключенных агентов.
- Параметр dY - **getConnectedAgent(index).getY() - getY()** - означает что мы размещаем точки каждой строки в точках подключенных агентов. Агенты поочередно общаются используя **getConnectedAgent(index)**, те элементы которые общаются будут представлены на презентации.
- Цвет линии - **getConnectedAgent(index).hashCode() > hashCode() ? new Color(226, 176, 153, 100) : null** – означает что мы меняем цвет, в зависимости от общения агентов.

Настроим элемент environment:

Тип расположения и тип сети – «Задается пользователем»

Добавим 4 переменные:

1. connectionsPerAgent – используется для подсчета соединений между агентами.

- Тип: double

- Начальное значение: 2

2. connectionRange – используется для подсчета дальних связей.

- Тип: double

- Начальное значение: 50

3. neighborLinkProbability - используется для подсчета вероятного соседа.

- Тип: double

- Начальное значение: 0.95

4. scaleFreeM – используется для подсчета свободно расположенных агентов.

- Тип: double

- Начальное значение: 10

Добавим 2 функции:

1. applyNetwork – функция используется для проверки и выполнения действий в переключателях в зависимости от настроек пользователя.

- Тип: void

- Код, тело функции:

```
switch( radioNetworkType.getValue() ) {
```

```
case 0:
```

```
environment.setNetworkUserDefined(); // устанавливает определенную сеть
```

```
break;
```

```
case 1:
```



```

environment.setNetworkRandom( connectionsPerAgent ); //устанавливает случайную сеть
break;
case 2:
environment.setNetworkAllInRange( connectionRange );// устанавливает сеть с диапазо-
ном
break;
case 3:
environment.setNetworkRingLattice( (int)connectionsPerAgent );// устанавливает кольце-
вую сеть
break;
case 4:
environment.setNetworkSmallWorld( (int)connectionsPerAgent, neighborLinkProbability
);//устанавливает малую сеть
break;
case 5:
environment.setNetworkScaleFree( scaleFreeM );// устанавливает свободную сеть
break;
}
environment.applyNetwork(); //выполняем функцию applyNetwork();//

```

2. setNumberOfAgents – функция используется для установки числа аген-
тов.

- Тип: void
- Код, тело функции:

```

while( comp.size() > n ) //до тех пор пока, число агентов больше заданного числа в пере-
ключателе radio,
remove_comp( comp.get(0) );// мы удаляем агенты.
while( comp.size() < n ) {
//до тех пор пока, число агентов меньше заданного числа в переключателе radio,
Agent a = add_comp();// мы добавляем агенты
a.setEnvironment( environment );
}
environment.applyNetwork(); //выполняем функцию applyNetwork();
environment.applyLayout(); // выполняем функцию applyLayout();

```

Добавим 3 переключателя из палитры «Элементы управления»:

1. radioNetworkType – используется для выбора типа сети.

Настройте его следующим образом:

Рисунок 8 – Настройка переключателя

Подпишем его, перенесем из палитры «Презентация» элемент текст. В свойствах «Текст» добавим следующее: **Network type**.

Строка **applyNetwork()**; - применить к сети.

2. radioLayoutType – используется для выбора макета сети.

Настройте его следующим образом:

Рисунок 9 – Настройка переключателя.

Подпишем его, перенесем из палитры «Презентация» элемент текст. В свойствах «Текст» добавим следующее: **Layout type**.

3. radio – используется для определения числа агентов

Рисунок 10 – Настройка переключателя.

Код действия:

```

switch( value ) {
case 0: setNumberOfAgents( 10 ); break;
case 1: setNumberOfAgents( 50 ); break;
case 2: setNumberOfAgents( 100 ); break;
case 3: setNumberOfAgents( 200 ); break;
case 4: setNumberOfAgents( 500 ); break;
}

```

Данный код представляет собой функцию выбора в переключателе, если мы выбираем первый элемент, то он распознает его как 0 и устанавливает количество агентов равным 10 и т. д.

Добавим 4 бегунка:

1. sliderConnectionsPerAgent – используется для определения подключений на одного агента.

Строка **radioNetworkType.getValue() == 1 || radioNetworkType.getValue() == 3 || radioNetworkType.getValue() == 4**, означает что мы сравниваем значение которое берем из первого переключателя с 1, 3, 4, проверяя тем самым доступность элемента.

Строка **applyNetwork();** - означает применение к сети, в случае, если есть доступность.

Подпишем его, перенесем из палитры «Презентация» элемент текст. В свойствах «Текст» добавим следующее: **Connections per agent:**

2. sliderRange – используется для установления диапазона присоединений.

Строка **radioNetworkType.getValue() == 2**, означает, что мы значение из первого переключателя сравниваем с числом 2, и выполняем, если совпадает.

Подпишем его, перенесем из палитры «Презентация» элемент текст. В свойствах «Текст» добавим следующее: **Connection range:**

3. sliderNeighborLinkProb – используется для установления диапазона соединений

Строка **radioNetworkType.getValue() == 4**, означает, что мы значение из первого переключателя сравниваем с числом 4, то есть проверяем, что выбрано в переключателе.

Подпишем его, перенесем из палитры «Презентация» элемент текст. В свойствах «Текст» добавим следующее: **Neighbor link probability:**

4. sliderScaleFreeM – используется для установления свободного расположения.

Строка **radioNetworkType.getValue() == 5**, означает, что мы значение из первого переключателя сравниваем с числом 5.

Подпишем его, перенесем из палитры «Презентация» элемент текст. В свойствах «Текст» добавим следующее: **M (Scale Free parameter):**

Запустите модель, проверьте ее работоспособность.

Задание 2: Создайте модель с двумя типами агентов - защитники и злоумышленники. Для защитников - правило следить за злоумышленниками, которые попадают в определенный радиус и не пропускать их к условному центру. Защитник должен двигаться в сторону злоумышленника, и если расстояние сократилось до установленного значения, то исключать агента-злоумышленника из дальнейшего процесса. Злоумышленникам действовать по правилам - двигаться к центру и изменять траекторию движения, если в определенном радиусе появился защитник.

Создайте модель с нуля.

Построим схему территории по которой будут перемещаться наши злоумышленники, защитники и обычные люди. (Рисунок 16)

Для построения используйте элемент прямоугольник из палитры «Презентация» и элементы библиотеки «3D Объекты».

Создайте улицы, по которым будут двигаться агенты, объедините их в группу – streets.

Далее создадим 3 агента: defender, malefactor, persons.

defender, malefactor при создании агентов поместите их в одну среду.

Настроим элемент defender:

- Начальное количество объектов – 3.

Настроим элемент malefactor:

- Начальное количество объектов – 10

Агент persons, создадим с отдельной средой.

- Начальное количество объектов – 100

Настроим среду для первых двух элементов:

В классе Main создадим 3 параметра:

1. Параметр meter – отвечает за радиус и скорость перемещения:

- Тип – double
- Значение по умолчанию – 5

2. scanRadius – переменная отвечает за радиус сканирования, обнаружения злоумышленника.

- Тип – double
- Значение по умолчанию - 10 * meter

3. killRadius – переменная отвечает на каком расстоянии от злоумышленника должен находиться защитник, что бы его ликвидировать.

- Тип – double
- Значение по умолчанию - 2 * meter

Далее создадим 2 функции: placeMeOnStreet, chooseNextDestinationForMe.

1. Функция chooseNextDestinationForMe – используется для нахождения злоумышленника.

Тело функции:

```
double x = agent.getX(); //получаем значение x и y агентов
double y = agent.getY();
ArrayList segments = new ArrayList();
for( int i=0; i<="" b="">запускаем цикл для поиска координат злоумышленника
ShapeRectangle rect = (ShapeRectangle)streets.get(i);
if( rect.contains( x, y ) )
segments.add( rect );
}
//выбираем сегмент улицы
int index = uniform_discr( segments.size()-1 );
ShapeRectangle rect = segments.get( index );
x = rect.getX() + uniform( rect.getWidth() );//получаем координаты
y = rect.getY() + uniform( rect.getHeight() );
//начинаем движение в нужном направлении
agent.moveTo( x, y );
```

2. Функция placeMeOnStreet – используется в начале работы программ для поиска каждого агента.

Тело функции:

```

double x = 0;
double y = 0; //обнуляем переменные
boolean ok = false;
while( ! ok ) {
x = uniform( 700 ); //присваиваем переменным значения области.
y = uniform( 500 );
for( int i=0; i<="" b="">цикл для поиска агентов в области
ShapeRectangle rect = (ShapeRectangle)streets.get(i);
if( rect.contains(x,y) ) {
ok = true;
break;
}
}
}
agent.jumpTo( x, y );

```

Настроим класс Defender:

Создадим 6 параметров:

1. meter – отвечает за радиус и скорость перемещения:

- Тип – double
- 2. scanRadius – радиус сканирования
- Тип – double
- Значение по умолчанию - **10 * meter**

3. killRadius - переменная отвечает на каком расстоянии от злоумышленника должен находиться защитник, что бы его ликвидировать.

- Тип – double
- Значение по умолчанию - **2 * meter**
- 4. velocityNormal – параметр отвечающий за нормальную скорость
- Тип – double
- Значение по умолчанию - **3 * meter / second()**

5. velocityFollowVictim – скорость преследования защитником нарушителей.

- Тип – double
- Значение по умолчанию - **3.3 * meter / second()**

6. decisionProbability – параметр отвечает за вероятное решение.

- Тип – double
- Значение по умолчанию - **0.2**

Добавим переменную victim, для подсчета количества злоумышленников.

- Тип, другой – Malefactor.

Добавим функцию - otherPeopleAround – отвечает за всех нарушителей в области:

Тело функции:

```

for( Malefactor v : get_Main().malefactor ) { //проходим через всех нарушителей в области
if( v.statechart.isActive( Malefactor.Killed ) )
continue; //пропускаем других людей
if( v != victim && distanceTo( v ) < scanRadius ) return true;
} //пропускаем нарушителей находящихся все зоны обнаружения
return false;

```

Начало диаграммы состояний – statechart

- Действие - **get_Main().placeMeOnStreet(this);**
- Состояние – Scanning
- Действие при входе - **get_Main().chooseNextDestinationForMe(this);** - движемся в любом направлении
- Переход – Scan

- Происходит – по таймауту - **0.1 * second()**
- Действие

```

victim = null; //сброс количества нарушителей
for( Malefactor v : get_Main().malefactor ) { //проходим через все жертвы в области
    if( v.statechart.isStateActive( Malefactor.Killed ) )
        continue; //пропускаем ликвидированных людей
    if( distanceTo( v ) < scanRadius ) { //если объект в пределах радиуса сканиро-
        вания
            if( randomTrue( decisionProbability ) ) { //случайный выбор
                victim = v; //запоминаем нарушителя
                break; //прекращаем поиск
            }
        }
    }

```

Переход – ChangeDirection

- Происходит - по прибытию агента.

Переход VictimChosen

- Происходит – при выполнении условия - **victim != null**

Состояние – Chasing

- Действие при входе - **setVelocity(velocityFollowVictim);** - увеличиваем скорость
- Действие при выходе - **setVelocity(velocityNormal);** - замедляем, возвращаем нормальную скорость

Переход – StepChase

- Происходит – по таймауту - **0.1 * second()**
- Действие - **moveTo(victim.getX(), victim.getY());** - двигаемся в сторону злоумышленника.

Переход – CloseEnoughToKill

- Происходит – при выполнении условия - **distanceTo(victim) < killRadius**, если радиус ликвидации больше, чем дистанция до нарушителя.

Состояние – Following, не требует настройки

Переход StepFollow

- Происходит – по таймауту - **0.1 * second()**
- Действие - **moveTo(victim.getX(), victim.getY());** , двигаемся в сторону нарушителя.

Переход NoOneAround

- Происходит - при выполнении условия - **! otherPeopleAround()** – поблизости нет других нарушителей.

Состояние – Killing

- Действие при входе - **stop();** - останавливаемся, **victim.statechart.receiveMessage("I killed you!");** - посылаем нарушителю сообщение, что он убит.

Переход – GiveUp

- Происходит – по таймауту - **15 * second()**
- Действие - **victim = null;** - отказываемся от преследования.

Переход – Completed

- Происходит – по таймауту - **10 * second()**
- Действие - **victim = null;** - прекращаем преследование.

Состояние – WalkAway

- Действие при входе - **get_Main().chooseNextDestinationForMe(this);** - начинаем движение в любом направлении.

Переход – ResumeScanning

- Происходит – по прибытию агента

Изменим агент класса Defender
Добавим элемент овал, из палитры «Презентация», настроим его следующим образом:

- Цвет заливки – **yellow**, прозрачность 59.
- Цвет линии – **gold**.

Добавим следующий овал:

- Цвет заливки – **red**, прозрачность 111.
- Цвет линии – **red**.

Третий овал:

- Цвет заливки – **mediumSlateBlue**, прозрачность 127.
- Цвет линии – **mediumSlateBlue**.

Линия, для соединения нарушителя с защитником.

- Цвет линии – **red**.

Далее настроим класс Person:
Начало диаграммы – statechart

- Действие - **get_Main().placeMeOnStreet(this);**
Состояние – Walking
- Действие при входе - **get_Main().chooseNextDestinationForMe(this);** - движемся в любом направлении.
Переход – ChangeDirection
- Происходит – по прибытию агента.
Настроим класс Malefactor:
Начало диаграммы – statechart
- Действие - **get_Main().placeMeOnStreet(this);**
Состояние – Walking
- Действие при входе - **get_Main().chooseNextDestinationForMe(this);** - движемся в любом направлении.
Переход – ChangeDirection
- Происходит – по прибытию агента.
Переход – transition
- Происходит – при получении сообщения.
- Тип сообщения – string
- Осуществлять переход, если сообщение равно: **"I killed you!"**
Состояние – Killed
- Действие при входе: **stop(); , groupPerson.setRotationY(PI/2)** – прекращаем движение.

Модель агента выглядит следующим образом:
Для реализации этой модели выберите из палитры «3D Объекты», картинку «человек», поместите его в прямоугольник:

- Цвет заливки – нет
- Цвет линии – red

Красное пятно можно сделать при помощи ломаной линии.
Добавим параметр meter

- Тип – double

В свойствах класса Defenders, Malefactor, во вкладке «Агент», зададим скорость равной - **3 * meter / second()**.

Запустите модель, проверьте работоспособность.

Лабораторная работа №7

Конечные автоматы

Рассмотрим пример покупки товаров, если имеются деньги. Построим ДКА (Детерминированный конечный автомат), с помощью автомата будем проверять, есть ли

деньги на покупку или нет. Если есть деньги, то мы покупаем товар до тех пор, пока не закончатся деньги.

Построим следующую схему.

Добавим три переменные:

1. Переменная **v**, необходима для генерации денег, если сгенерированное число больше нуля, то у нас есть деньги на покупку товаров. (Тип – int, начальное значение – 0)
2. Переменная **symbol_**, вспомогательная переменная, если переменная **v** больше нуля, то переменной присваиваем значение 1, а если нет, то 0. Служит для более простой реализации генерации входных чисел. (Тип – int, начальное значение – 0)
3. Переменная **count_food**, необходима для подсчета купленной еды. (Тип – int, начальное значение – 0)

Настроим элементы следующим образом:

Элемент	Свойство	Настройка
Состояние q0	Действие при входе	<code>if(v>0)//если переменная больше нуля, { qO.setFillColor(green);//то круг q0 - зеленый symbol_=1;//переменная равна единице. } else //иначе { qO.setFillColor(red);//красный symbol_=0;//переменная равна 0 }</code>
	Действие при выходе	<code>qO.setFillColor(gray);</code>
Переход проверка_наличия_денег	Происходит при выполнении условия	<code>symbol_==0;</code>
Переход если_есть_деньги	Происходит при выполнении условия	<code>symbol_==1;</code>
Состояние пустой_блок1		Служит для наглядности совершения покупок.

Переход задерж- ка_времени_ для_наглядности_ работы1	Происходит по таймауту	2
Состояние q1	Действие при входе	if (v<=0) { symbol_=0; } else { symbol_=1; qO1.setFillColor(green); }
	Действие при выходе	qO1.setFillColor(gray);
Переход задерж- ка_времени_ для_наглядности_ работы2	Происходит по таймауту	0,5
Состояние пустой_блок2	Служит для наглядности совершения покупок.	
Переход покупка	Происходит при выпол- нении усло- вия	symbol_==1;
	Действие	v=v-1;//отнимаем за каждую покупку count_food=count_food+1;//прибавляем товар за каждую покупку
Переход если_деньги_ кончились	Происходит при выпол- нении усло- вия	symbol_==0;
Состояние q2	Действие при входе	qO2.setFillColor(green);
	Действие при выходе	qO2.setFillColor(gray);
Конечное состояние finalState	Действие	qO.setFillColor(red); qO1.setFillColor(red); qO2.setFillColor(red);

Далее добавим кнопку, для запуска модели. Настроим ее следующим образом:
Основные свойства, действие:


```

v = (int)uniform_discr( 1, 100 );
statechart.start();
qO.setFillColor( gray );
qO1.setFillColor( gray );
qO2.setFillColor( gray );

```

Добавим в Main, свойство «Действие при запуске», следующий код:

```
v = (int)uniform_discr( 1, 100 ); //генерация чисел от 1 до 100
```

Далее добавим для наглядности следующую модель:

Рисунок 6 – Модель покупки товаров.

Для построения модели использовались элементы палитры «Презентация».

Запустите модель, проверьте ее работоспособность,

Перечень вопросов для промежуточной аттестации на зачете (1 семестр для очно-заочной и заочной формы обучения):

1. Независимые испытания.
2. Формула полной вероятности.
3. Случайные величины.
4. Математическое ожидание случайных величин.
5. Независимость случайных величин.
6. Дисперсия случайных величин.
7. Среднеквадратичное отклонение.
8. Распределение случайной величины: определение понятия.
9. Стандартное нормальное распределение.
10. Правило трех сигм.
11. Центральная предельная теорема.
12. Экспоненциальное распределение.
13. Распределения. Вейбулла-Гнеденко.
14. Дискретные распределения: биномиальные, гипергеометрические и Пуассона.
15. Статистические методы обработки экспериментальных данных.
16. Статистические оценки.
17. Несмещенные, эффективные и состоятельные оценки.
18. Выборочная средняя и выборочная дисперсия.
19. Число степеней свободы.
20. Основные законы распределения статистических оценок. Точечная и интервальная оценки. Доверительный интервал. Метод моментов и метод наибольшего правдоподобия для точечной оценки параметров распределения.
21. Статистическая гипотеза. Проверка статистических гипотез. Ошибки первого и второго рода. Статистический критерий проверки нулевой гипотезы.
22. Проверка гипотезы о распределении генеральной совокупности.
23. Критерий Пирсона.
24. Зависимости между случайными величинами.
25. Типы зависимостей.
26. Линейная связь.
27. Корреляция.
28. Регрессионный анализ.
29. Выборочное уравнение регрессии.
30. Отыскание параметров выборочного уравнения регрессии по несгруппированным и сгруппированным данным.
31. Дисперсионный анализ.
32. Понятие о дисперсионном анализе.
33. Факторная и остаточная дисперсии.

34. Основные понятия многомерного статистического анализа.
35. Методы факторного анализа, их область применения.
36. Метод главных компонент.
37. Классификация объектов, описываемых количественными и качественными признаками.
38. Примеры кластер-анализа.
39. Понятие модели.
40. Классификация моделей.
41. Понятие моделирования.
42. Виды моделирования.
43. Этапы моделирования.
44. Назначение этапов моделирования.
45. Стадии исследования.
46. Эксперимент.
47. Виды экспериментов.
48. Основные задачи, решаемые планированием эксперимента.

Перечень вопросов для промежуточной аттестации на экзамене (1 семестр для очной формы обучения, 2 семестр для очно-заочной и заочной формы обучения):

1. Независимые испытания.
2. Формула полной вероятности.
3. Случайные величины. Математическое ожидание случайных величин.
4. Независимость случайных величин. Дисперсия случайных величин.
5. Среднеквадратичное отклонение.
6. Распределение случайной величины: определение понятия. Стандартное нормальное распределение.
7. Правило трех сигм.
8. Центральная предельная теорема.
9. Экспоненциальное распределение.
10. Распределения. Вейбулла-Гнеденко.
11. Дискретные распределения: биномиальные, гипергеометрические и Пуассона.
12. Статистические методы обработки экспериментальных данных. Статистические оценки.
13. Несмещенные, эффективные и состоятельные оценки.
14. Выборочная средняя и выборочная дисперсия.
15. Число степеней свободы.
16. Основные законы распределения статистических оценок. Точечная и интервальная оценки. Доверительный интервал.
17. Метод моментов и метод наибольшего правдоподобия для точечной оценки параметров распределения.
18. Статистическая гипотеза. Проверка статистических гипотез. Ошибки первого и второго рода. Статистический критерий проверки нулевой гипотезы.
19. Проверка гипотезы о распределении генеральной совокупности.
20. Критерий Пирсона.
21. Зависимости между случайными величинами. Типы зависимостей.
22. Линейная связь.
23. Корреляция.
24. Регрессионный анализ. Выборочное уравнение регрессии.
25. Отыскание параметров выборочного уравнения регрессии по несгруппированным и сгруппированным данным.
26. Дисперсионный анализ. Понятие о дисперсионном анализе.

27. Факторная и остаточная дисперсии.
28. Основные понятия многомерного статистического анализа.
- 29.
30. Методы факторного анализа, их область применения.
31. Метод главных компонент.
32. Классификация объектов, описываемых количественными и качественными признаками.
33. Примеры кластер-анализа.
34. Понятие модели. Классификация моделей.
35. Понятие моделирования. Виды моделирования.
36. Этапы моделирования. Назначение этапов моделирования.
37. Стадии исследования.
38. Эксперимент. Виды экспериментов.
39. Основные задачи, решаемые планированием эксперимента
40. Измерения. Виды измерений.
41. Класс точности измерительных приборов. Абсолютная и относительная погрешность.
42. Ошибки наблюдения и их виды (систематические, случайные, грубые).
43. Выбросы. Правила округления чисел. Значение цифры числа.
44. Доверительное оценивание: доверительная вероятность, доверительный интервал, надежность.
45. Взвешенное среднее.
46. Обработка результатов прямых и косвенных измерений.
47. Выборочная совокупность. Репрезентативность выборки. Виды выборок.
48. . Дисперсия.
49. . Распределение вероятностей Стюдента.
50. . Проверка гипотез: ошибки первого, второго рода.
51. . Проверка однородности независимых выборок.
52. . Проверка воспроизводимости опытов.
53. . Критерий Стюдента, критерий Фишера, критерий Кохрена.
54. . Выявление промахов при обработке результатов измерений. Критерий Шовене.
55. . Понятие регрессионной зависимости. Уравнение регрессии.
56. . Понятие корреляции.
57. . Основная задача регрессионного анализа.
58. . Основная задача корреляционного анализа.
59. . Однофакторная линейная регрессия.
60. . Определение коэффициентов регрессии линейной однофакторной зависимости методом наименьших квадратов.
61. . Выборочный коэффициент корреляции. Свойства коэффициента корреляции. Оценка значимости коэффициента корреляции.
62. . Множественная линейная регрессия.
63. . Полиномиальная регрессия.
64. . Проверка регрессии на достоверность и адекватность.
65. . Дисперсия во воспроизводимости.
66. . Анализ остатков.
67. . Выделение факторов, оказывающих наибольшее влияние на отклик. Оценка характера, знака и степени влияния факторов на отклик.
68. . Прогноз значения отклика для любых значений факторов.
69. . Задача оптимизации.
70. . Метод крутого восхождения по поверхности отклика.

5.5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности, характеризующих этапы формирования компетенций

Критерии оценки ответа на вопросы зачета.

Ответы обучающегося на зачете оцениваются педагогическим работником по варианту зачтено/незачтено.

«Зачтено» – обучающийся глубоко и прочно освоил программный материал, исчерпывающе, последовательно, грамотно и логически стройно его излагает, тесно увязывает с задачами и будущей деятельностью, не затрудняется с ответом при видоизменении задания, свободно справляется с задачами и практическими заданиями, правильно обосновывает принятые решения, умеет самостоятельно обобщать и излагать материал, не допуская ошибок;

«Незачтено» - обучающийся не знает значительной части программного материала, допускает существенные ошибки, с большими затруднениями выполняет практические задания, задачи.

Критерии оценки ответа на вопросы экзамена:

Ответы обучающегося **на экзамене** оцениваются педагогическим работником по **пятибалльной системе**.

5 баллов – обучающийся глубоко и прочно освоил программный материал, исчерпывающе, последовательно, грамотно и логически стройно его излагает, тесно увязывает с задачами и будущей деятельностью, не затрудняется с ответом при видоизменении задания, свободно справляется с задачами и практическими заданиями, правильно обосновывает принятые решения, умеет самостоятельно обобщать и излагать материал, не допуская ошибок;

4 баллов – обучающийся твердо знает программный материал, грамотно и по существу излагает его, не допуская существенных неточностей в ответе на вопрос, может правильно применять теоретические положения и владеет необходимыми умениями и навыками при выполнении практических заданий;

3 баллов – обучающийся освоил основной материал, но не знает отдельных деталей, допускает неточности, недостаточно правильные формулировки, нарушает последовательность в изложении программного материала и испытывает затруднения в выполнении практических заданий;

2 баллов – обучающийся не знает значительной части программного материала, допускает существенные ошибки, с большими затруднениями выполняет практические задания, задачи.

6. Перечень основной и дополнительной учебной литературы для освоения учебной дисциплины

6.1. Основная литература.

1. Математическое моделирование. Практикум: учебное пособие / Л.А. Коробова, Ю.В. Бугаев, С.Н. Черняева, Ю.А. Сафонова; науч. ред. Л.А. Коробова; Воронежский государственный университет инженерных технологий. – Воронеж: Воронежский государственный университет инженерных технологий, 2017. – 113 с.: Режим доступа: по подписке. – URL: <http://biblioclub.ru/index.php?page=book&id=482006>

2. Шорохова, И.С. Статистические методы анализа: учебное пособие / И.С. Шорохова, Н.В. Кисляк, О.С. Мариев; Уральский федеральный университет им. первого Президента России Б. Н. Ельцина. – 2-е изд., стер. – Москва: Флинта: Уральский федеральный университет (УрФУ), 2017. – 301 с.: Режим доступа: по подписке. – URL: <http://biblioclub.ru/index.php?page=book&id=482354>

6.2. Дополнительная литература

1. Семенов, А.Г. Математическое и компьютерное моделирование: практикум: [16+] / А.Г. Семенов, И.А. Печерских; Кемеровский государственный университет. – Кемерово: Кемеровский государственный университет, 2019. – 237 с : Режим доступа: по подписке. – URL: <http://biblioclub.ru/index.php?page=book&id=574121>

7. Перечень ресурсов информационно-телекоммуникационной сети «Интернет», современные профессиональные базы данных и информационные системы необходимые для освоения учебной дисциплины

1. <http://mon.gov.ru> – сайт Минобрнауки РФ
2. <http://www.edu.ru/> – библиотека федерального портала «Российское образование» (содержит каталог ссылок на интернет-ресурсы, электронные библиотеки по различным вопросам образования)
3. <http://www.prilib.ru> – Президентская библиотека
4. <http://www.rusneb.ru> – Национальная электронная библиотека
5. <http://elibrary.rsl.ru/> – сайт Российской государственной библиотеки (раздел «Электронная библиотека»)
6. <http://elibrary.ru> – научная электронная библиотека «Elibrary»
7. <https://uisrussia.msu.ru> - университетская информационная система РОССИЯ (УИС РОССИЯ)

8. Методические указания для обучающихся по освоению учебной дисциплины

Освоение обучающимся учебной дисциплины **«Математическое моделирование и статистическая обработка данных»** предполагает изучение материалов дисциплины на аудиторных занятиях и в ходе самостоятельной работы. Аудиторные занятия проходят в форме лекций и практических занятий. Самостоятельная работа включает разнообразный комплекс видов и форм работы обучающихся.

Для успешного освоения учебной дисциплины и достижения поставленных целей необходимо внимательно ознакомиться настоящей рабочей программы учебной дисциплины. Ее может представить преподаватель на вводной лекции или самостоятельно обучающийся использует информацию на официальном Интернет-сайте Института.

Следует обратить внимание на список основной и дополнительной литературы, на предлагаемые преподавателем ресурсы информационно-телекоммуникационной сети Интернет. Эта информация необходима для самостоятельной работы обучающегося.

При подготовке к аудиторным занятиям необходимо помнить особенности каждой формы его проведения.

Подготовка к учебному занятию лекционного типа заключается в следующем.

С целью обеспечения успешного обучения обучающийся должен готовиться к лекции, поскольку она является важнейшей формой организации учебного процесса, поскольку:

- знакомит с новым учебным материалом;
- разъясняет учебные элементы, трудные для понимания;
- систематизирует учебный материал;
- ориентирует в учебном процессе.

С этой целью:

- внимательно прочитайте материал предыдущей лекции;
- ознакомьтесь с учебным материалом по учебнику и учебным пособиям с темой прочитанной лекции;
- внесите дополнения к полученным ранее знаниям по теме лекции на полях лекционной тетради;

- запишите возможные вопросы, которые вы зададите лектору на лекции по материалу изученной лекции;
- постарайтесь уяснить место изучаемой темы в своей подготовке;
- узнайте тему предстоящей лекции (по тематическому плану, по информации лектора) и запишите информацию, которой вы владеете по данному вопросу

Подготовка к практическому занятию

При подготовке к практическому занятию следует обратить внимание на следующие моменты: на процесс предварительной подготовки, на работу во время занятия, обработку полученных результатов, исправление полученных замечаний.

Предварительная подготовка к практическому занятию заключается в изучении теоретического материала в отведенное для самостоятельной работы время, ознакомление с инструктивными материалами с целью осознания задач практического занятия.

Работа во время проведения практического занятия включает несколько моментов:

- консультирование студентов преподавателями с целью предоставления исчерпывающей информации, необходимой для самостоятельного выполнения предложенных преподавателем задач, ознакомление с правилами техники безопасности при работе в аудитории;
- самостоятельное выполнение заданий согласно обозначенной учебной программой тематики.

Самостоятельная работа.

Для более углубленного изучения темы задания для самостоятельной работы рекомендуется выполнять параллельно с изучением данной темы. При выполнении заданий по возможности используйте наглядное представление материала.

Подготовка к зачету с оценкой.

К зачету с оценкой необходимо готовиться целенаправленно, регулярно, систематически и с первых дней обучения по данной дисциплине.

При подготовке к зачету с оценкой обратите внимание на практические задания на основе теоретического материала.

При подготовке к ответу на вопросы зачета с оценкой по теоретической части учебной дисциплины выделите в вопросе главное, существенное (понятия, признаки, классификации и пр.), приведите примеры, иллюстрирующие теоретические положения.

После предложенных указаний у обучающихся должно сформироваться четкое представление об объеме и характере знаний и умений, которыми надо будет овладеть по дисциплине.

9. Программное обеспечение информационно-коммуникационных технологий

9.1. Информационные технологии

1. Персональные компьютеры
2. Доступ к интернет
3. Проектор
4. Система VOTUM

9.2. Программное обеспечение

1. Windows 7

2. LibreOffice Writer,
3. LibreOffice Calc,
4. LibreOffice Impress

9.3. Информационные справочные системы

1. Университетская информационная система РОССИЯ - <http://www.cir.ru/>

10. Перечень материально-технического обеспечения, необходимого для реализации программы по учебной дисциплине

1. Учебные аудитории для проведения занятий лекционного типа, практических занятий, групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации.
2. Помещения для самостоятельной работы студентов, оснащенные компьютерной техникой с возможностью подключения к сети «Интернет» и обеспечением доступа в электронную информационно-образовательную среду института.
3. Лаборатории, оснащенные лабораторным оборудованием: персональные компьютеры.

11. Образовательные технологии

При реализации учебной дисциплины **«Математическое моделирование и статистическая обработка данных»** применяются различные образовательные технологии.

Освоение учебной дисциплины **«Математическое моделирование и статистическая обработка данных»** предусматривает использование в учебном процессе активных форм проведения учебных занятий в форме, разбор конкретных ситуаций и практических задач в сочетании с внеаудиторной работой с целью формирования и развития профессиональных навыков обучающихся.