

**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО
ОБРАЗОВАНИЯ, САНКТ-ПЕТЕРБУРГСКИЙ НАЦИОНАЛЬНЫЙ
ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИНФОРМАЦИОННЫХ
ТЕХНОЛОГИЙ, МЕХАНИКИ И ОПТИКИ**

Факультет Компьютерных технологий и управления

Кафедра Вычислительной техники

Направление подготовки (специальность): 09.03.04 - «Программная инженерия»

ОТЧЕТ

по практике

Тема задания:

Студент: Игнатъева Ю.В. **Группа:** Р3311

Руководитель практики: Соснин Владимир Валерьевич

Оценка, рекомендуемая руководителем: _____

Практика пройдена с оценкой : _____

Подписи членов комиссии

_____ (_____)

_____ (_____)

_____ (_____)

Дата _____

Санкт-Петербург

2016

Оглавление

1	Система компьютерной вёрстки TeX (LaTeX)	2
1.1	Краткое описание LaTeX	2
1.2	Перечень сильных и слабых сторон TeX	3
1.3	Инструмент для редактирования TeX - TeXworks	4
2	Система контроля версий Git	5
2.1	Краткое описание Git	5
2.2	Описание сильных и слабых сторон Git	5
3	Литература	7
3.1	А.Г. Куприяшкин	7
3.2	В.Д. Боев	9
4	Выполнение индивидуального задания	10
4.1	Описание моделей СМО	10
4.2	Создание моделей СМО в GPSS	10
4.3	Создание моделей СМО в AnyLogic	12
4.4	Замеры времени моделирования в GPSS	13
4.5	Замеры времени моделирования в AnyLogic	13
5	Выводы по работе	15

Глава 1

Система компьютерной вёрстки TeX (LaTeX)

1.1 Краткое описание LaTeX

В 1978 году профессор Гарвардского университета Дональд Кнут опубликовал первую версию систему для верстки текстов, которая ныне известна как TeX. Кнут писал TeX во времена появления цифрового печатающего оборудования, чтобы изучить его потенциал и, в особенности, обратить тенденцию ухудшения типографского качества, которую он наблюдал на примере его собственных книг. TeX произносится как "тех". Последняя буква X в названии TeX — вовсе не английская буква "икс" (x), а греческая "хи". TeX считается наиболее качественной системой подготовки текстов. Как сказано в словаре компьютерных терминов (Douglas Downing and Michael A. Covington. Dictionary of Computer and Internet Terms. 8th ed. Barron's Educational Series. New York: Woodbury, 2003.) TeX определяет стандарт, к которому пытаются приблизиться другие настольные издательские системы.

В начале восьмидесятых годов XX века Лесли Лампорт разработал систему LaTeX, которая была основана на форматирующих средствах TeX'a. LaTeX - макропакет, который позволяет при помощи уже заранее определенных, профессиональных макетов верстать авторам их работы с высоким типографским качеством. LaTeX произносится как "латех". Чтобы пользоваться системой LaTeX и создавать удобные для чтения текстовые произведения, совсем не надо быть умелым пользователем системы TeX - достаточно выбрать готовый стиль и использовать несколько простых команд в зависимости от того, что нужно в данном случае. Набор макросов, для решения задач, пополнялся с каждым годом, поэтому в 1992 году был организован файловый архив CTAN. CTAN - это акроним "Comprehensive TeX Archive Network". Будучи распространяемым под лицензией LaTeX Project Public License, LaTeX относится к свободному программному обеспечению.

Главная идея LaTeX состоит в том, что теперь пользователь может сконцентрировать свои усилия на содержании и структуре текста, о том, что он напишет, не заботясь о деталях его оформления. Готовя свой документ, автор указывает логическую структуру текста (разбивая его на главы, разделы, таблицы, изображения), а LaTeX решает вопросы его отображения. Так содержание отделяется от оформления. Оформление при этом или определяется заранее (стандартное), или разрабатывается для конкретного документа. В конце девяностых годов XX века такая идея отделения содержания от оформления нашла своё продолжение в XML - расширяемом языке разметки (eXtensible Markup Language).

В России TeX и LaTeX появились в конце восьмидесятых годов, тогда же был разработан алгоритм автоматического переноса русских слов.

Пакет позволяет автоматизировать многие задачи набора текста и подготовки статей, включая

1. набор текста на нескольких языках;
2. нумерацию разделов и формул;
3. перекрёстные ссылки;
4. размещение иллюстраций и таблиц на странице;
5. генерацию оглавлений, списков иллюстраций и таблиц;
6. генерацию предметных указателей;
7. ведение библиографии и др.

1.2 Перечень сильных и слабых сторон TeX

Перечислим основные достоинства TeX:

1. LaTeX предоставляет удобные и гибкие средства, чтобы напечатанный вами текст был "совсем как в книге". В частности, указав с помощью простых средств логическую структуру текста, автор может не вникать в детали оформления. Но при необходимости, эти детали могут быть изменены;
2. высокое качество и гибкость верстки абзацев и математических формул;
3. TeX неприхотлив к используемой технике. Например, им можно пользоваться на компьютерах на базе 80286-процессора;
4. TeXовские файлы обладают высокой степенью переносимости. TeX чрезвычайно мобилен и свободно доступен, поэтому система работает практически на всех существующих платформах.

Однако, есть у TeXа и недостатки:

1. TeX работает относительно медленно, занимает много памяти;
2. некоторые пользователи, которые привыкли работать с редакторами наподобие Chiwriter, относят к недостаткам TeXа тот факт, что он не является системой типа WYSIWYG, то есть работа с исходным текстом и просмотр того, как текст будет выглядеть на печати, - разные операции. Но с другой стороны, благодаря этой особенности TeX время на подготовку текста сокращается;
3. переносимость TeXовских текстов снижается, если в них предусмотрен импорт графических файлов;
4. хотя предопределенные макеты имеют множество настраиваемых параметров, создание нового макета документа не очень простое дело и может занять много времени;
5. очень сложно писать неструктурированные и неорганизованные документы.

Не смотря на то, что недостатков получилось больше, преимущества TeX позволяют ему быть популярным среди пользователей, существовать для всех типов компьютеров и применяться для подготовки как одностраничных писем, так и многотомных фолиантов. Многие редакции научных журналов рекомендуют, а иногда и вынуждают готовить статьи в системе LaTeX, чтобы потом, заменив всего лишь одно слово в названии класса печатного документа в преамбуле входного файла, издатель мог придать тексту тот облик, который отличает выбранный журнал.

1.3 Инструмент для редактирования TeX - TeXworks

Для редактирования отчета мной была выбрана свободная среда для работы с TeX-документами, включающая редактор, просмотрщик PDF - TeXworks, обладающая простым интерфейсом. TeXworks является удобным в использовании графическим интерфейсом (GUI) к системе подготовки документов TeX/LaTeX и других. Разработчик данного редактора Джонатан Кью. В комплект TeXworks включен набор шаблонов для создания наиболее часто используемых документов. Редактор включает в себя базовые возможности (характерные для большинства редакторов текста), систему автодополнения команд, запуск вёрстки и возможность добавления сценариев для преобразования текста документа.

Этот редактор был мной выбран, так как он идеально подходит при необходимости получить PDF файл в результате вёрстки (используется pdfTeX). Для предварительного просмотра скомпилированных документов имеется интегрированный просмотрщик PDF (основанный на библиотеке Poppler), поддерживающий синхронизацию с исходным документом.

Глава 2

Система контроля версий Git

2.1 Краткое описание Git

Чтобы хранить и регистрировать изменения в файле с отчетом по практике, чтобы в дальнейшем была возможность вернуться к определённым старым версиям этого файла, мною была использована система контроля версий Git. Git (произносится как «гит») - распределённая система управления версиями, которая была создана Линусом Торвальдсом. Первая версия появилась в 2005 году. Очень много проектов используют Git, самые известные из них: ядро Linux, Android, MediaWiki, DokuWiki, некоторые дистрибутивы Linux и другие. Программа является свободной.

К особенностям реализации этой системы можно отнести то, что ядро Git представляет собой набор утилит командной строки с параметрами, а все настройки хранятся в текстовых файлах конфигурации. Такая реализация делает Git легко портируемым на любую платформу.

Репозиторий Git представляет собой каталог файловой системы, в котором находятся файлы конфигурации репозитория, файлы журналов, хранящие операции, выполняемые над репозиторием, индекс, описывающий расположение файлов и хранилище, содержащее собственно файлы.

2.2 Описание сильных и слабых сторон Git

Сравнение Git другими системами контроля версий можно разделить на два этапа.

Так как Git является распределённой системой контроля версий, в первую очередь сравним ее с "родственными" ей системами. Наиболее популярными и называемыми преимуществами Git перед другими системами являются:

1. высокая производительность;
2. продуманная система команд, позволяющая удобно встраивать git в скрипты;
3. качественный веб-интерфейс;
4. репозитории git могут распространяться и обновляться общесистемными файловыми утилитами архивации и обновления, такими как rsync;
5. Развитые средства интеграции с другими системами контроля версий, в частности, с CVS, SVN и Mercurial.

В числе недостатков Git обычно называют:

1. некоторое неудобство для пользователей, переходящих с других систем. Команды Git, ориентированные на наборы изменений, а не на файлы, могут вызвать недоумение у пользователей, привыкших к файл-ориентированным системам контроля версий, таким как SVN;

2. большие затраты времени, по сравнению с файл-ориентированными системами, на формирование истории конкретного файла, истории правок конкретного пользователя, поиска изменений, относящихся к заданному месту определённого файла;
3. большие накладные расходы при работе с проектами, в которых делаются многочисленные несвязанные между собой изменения файлов;
4. система работает только с файлами и их содержимым, и не отслеживает пустые каталоги.

Во-вторых, сравним систему Git с централизованными системами управления версиями (такими как, например, Subversion). Основные преимущества распределённых систем - их гибкость и значительно большая (по сравнению с централизованными системами) автономия отдельного рабочего места. Каждый компьютер разработчика является, фактически, самостоятельным и полнофункциональным сервером, из таких компьютеров можно построить произвольную по структуре и уровню сложности систему, задав (как техническими, так и административными мерами) желаемый порядок синхронизации. При этом каждый разработчик может вести работу независимо, так, как ему удобно, изменяя и сохраняя промежуточные версии документов, пользуясь всеми возможностями системы (в том числе доступом к истории изменений) даже в отсутствие сетевого соединения с сервером. Связь с сервером или другими разработчиками требуется исключительно для проведения синхронизации, при этом обмен наборами изменений может осуществляться по различным схемам.

К недостаткам распределённых систем можно отнести увеличение требуемого объёма дисковой памяти: на каждом компьютере приходится хранить полную историю версий, тогда как в централизованной системе на компьютере разработчика обычно хранится лишь рабочая копия, то есть срез репозитория на какой-то момент времени и внесённые изменения. Менее очевидным, но неприятным недостатком является то, что в распределённой системе практически невозможно реализовать некоторые виды функциональности, предоставляемые централизованными системами. Это:

1. блокировка файла или группы файлов (для хранения признака блокировки нужен общедоступный и постоянно находящийся в онлайне центральный сервер);
2. слежение за определённым файлом или группой файлов (изменения файлов происходят на разных серверах, слияния и выделения ветвей происходят локально, об изменениях становится известно только при синхронизации, причём не всем разработчикам, а только тем, кто в данной синхронизации участвует);
3. единая сквозная нумерация версий системы и/или файлов, в которой номер версии монотонно возрастает (такая нумерация также требует наличия главного сервера, задающего номера версий для всех остальных);
4. локальная работа пользователя с отдельной, небольшой по объёму выборкой из значительного по размеру и внутренней сложности хранилища на удалённом сервере.

С момента рождения в 2005 году Git развивался и эволюционировал, становясь проще и удобнее в использовании, сохраняя при этом свои первоначальные качества. Он невероятно быстр, очень эффективен для больших проектов, а также обладает превосходной системой ветвления для нелинейной разработки.

Глава 3

Литература

Цель практики - найти систему имитационного моделирования (СИМ), которая быстрее остальных умеет моделировать работы простейшей СМО М/М/1 с двумя видами приоритетов (ДОБП, ДО-ОП) для двух классов заявок. Но перед выполнением практической части задания, надо подробно изучить каждую СИМ, которая будет использоваться. Первичный список СИМ:

1. GPSS World;
2. Anylogic 7.1.

Остальные системы будут рассматриваться после сравнения вышеперечисленных СИМ. Для изучения системы AnyLogic мною была выбрана следующая литература:

1. А.Г. Куприяшкин "Основы моделирования систем". Учебное пособие посвящено моделированию динамических систем в среде AnyLogic. Показаны основные этапы разработки моделей – от постановки задачи до создания анимированной презентации.
2. В.Д. Боев "Концептуальное проектирование систем в Anylogic 7 и GPSS World". В этой книге предлагаются различные методики разработки имитационных моделей с применением инструментальных средств AnyLogic 7 и GPSS World.
3. В.Д. Боев "Компьютерное моделирование. Пособие для практических занятий, курсового и дипломного проектирования в AnyLogic 7". В этом пособии предлагаются методики разработки имитационных моделей проектируемых систем с применением инструментальных средств AnyLogic 7. Приводятся сравнительные оценки результатов моделирования разнородных дискретных процессов, полученных на моделях одной и той же проектируемой системы в GPSS World и AnyLogic 7.
4. Ilya Grigoryev "AnyLogic 7 in Three Days: A Quick Course in Simulation Modeling".

Для изучения GPSS World я буду использовать уже упомянутые книги В.Д.Боева, а так же:

1. Томашевский В.Н., Жданова Е.Т. "Имитационное моделирование в среде GPSS".
2. Алиев Т.И. "Основы моделирования дискретных систем".

Кратко перескажем основные моменты из прочитанных книг.

3.1 А.Г. Куприяшкин

Учебное пособие А.Г. Куприяшкина "Основы моделирования систем" содержит шесть разделов, материал которых разбит на параграфы.

В первом разделе расписаны основные понятия, которые могут вызывать затруднения у академической аудитории, например, что такое адекватность модели (это соответствие модели оригиналу, характеризующее степень близости свойств модели свойствам исследуемой системы), верификация (т.е. проверка достоверности модели), валидация (это процесс, позволяющий установить, является ли модель точным представлением системы для конкретных целей исследования) и т.д. Так же в этом разделе приведены методы для построения математических моделей, задачи моделирования, направления и инструменты моделирования. Вся информация этого раздела даст неподготовленному читателю представление о понятии моделирования и о его целях.

Во втором разделе "Основ моделирования систем" подробно описывается среда инструмент имитационного моделирования AnyLogic. Инструмент обладает современным графическим русскоязычным интерфейсом и позволяет использовать язык Java для разработки моделей. Автор подробно, начиная от регистрации на сайте компании и установки, показывает с помощью понятного материала и изображений этапы работы в данной среде, рассказывает про каждый элемент интерфейса, помогает создать наш собственный первый проект. В результате появляется представление о том, как надо планировать проект, для чего нужно формулировать его задачу, разделять его изначально на этапы.

В третьем разделе говорится о детерминированных непрерывных системах. В первом параграфе этого раздела автор приводит основные сведения, что учитывается при моделировании физических систем, описывает четыре режима (стационарный, переходный, периодический, динамический) динамики системы и объясняет все это на простой модели реактора идеального смешения. Далее автор предлагает нам в качестве примера построить имитационную модель пружинного маятника, эту задачу он делит на два этапа: математическую ("формализованную") и имитационную интерактивную ("неформализованную"). Затем добавляем в модель трехмерную анимацию. В AnyLogic она отображается в специальных окнах. В результате создалась интерактивная модель, оснащенная элементами 2D-3D-визуализации, средствами навигации и управления. Существует область задач, когда параметры системы изменяются по времени и пространственным координатам. Для таких систем создаются модели с распределенными параметрами, которые так же затрагиваются автором в этом разделе. В конце этого раздела А.Г. Куприяшкин формулирует обратную задачу моделирования, которая сводится к нахождению таких значений параметров модели, которые бы соответствовали определенному состоянию системы, и её решение.

Четвертый раздел посвящен дискретно-событийному моделированию. Здесь рассматриваются системы, которые обладают разнообразным поведением, и в зависимости от входных сигналов или событий переходят в разные состояния, дается понятие "конечный автомат" (математическая модель поведения устройств с конечной памятью), а так же приводится пример реализации по-настоящему культовой игры "Жизнь".

В последней главе автор пишет о моделировании систем массового обслуживания. В первую очередь, он дает определения основным понятиям, связанным с системами массового обслуживания (СМО), а затем показывает все на понятном примере с моделированием процесса выполнения задач компьютером.

В конце книги есть вопросы и задания на моделирование, где можно проверить понимание темы, а так же есть библиографический список с перечнем рекомендуемой автором, а так же используемой в книге, литературой. Эта книга предназначена как для академической, так и для профессиональной аудитории. В частности, опыт и знания, полученные при реализации в AnyLogic различных примеров моделей, помогут мне при сравнении Anylogic с другими системами имитационного моделирования.

3.2 В.Д. Боев

"Концептуальное проектирование систем в Anylogic 7 и GPSS World" написано в виде курса для дистанционного обучения. Курс предназначен для проведения практических занятий по дисциплинам, связанным с проектированием и компьютерным моделированием систем с использованием систем имитационного моделирования, и рассчитан на 16 часов.

Курс разбит на девять лекций, в каждой из которых рассматриваются пример модели, реализованный как в AnyLogic, так и в GPSS. Для первой лекции выбрана модель несложная и понятная многим: модель обработки запросов сервером. В последующих лекциях рассматриваются модели процесса изготовления в цехе деталей, функционирования направления связи, функционирования сети связи, функционирования системы связи, функционирования предприятия, предоставления ремонтных услуг, функционирования системы воздушных перевозок и обработки документов в организации. Автор подробно расписал все этапы, начиная с постановки задачи, а в первых лекциях даже сделал акценты на самих инструментах имитационного моделирования GPSS и AnyLogic для людей, которые работают с этими программами в первый раз. Последние лекции рассчитаны на людей, прошедших весь курс, либо на профессиональную аудиторию.

За этот день я проработала все эти модели, последние из них смоделировала в AnyLogic за исключением анимации, ведь по индивидуальному заданию модели в разных СИМ должны работать без визуализации моделирования.

Учебное пособие В.Д. Боева "Компьютерное моделирование. Пособие для практических занятий, курсового и дипломного проектирования в AnyLogic 7" содержит ту же информацию, но оформлено в виде книги.

Глава 4

Выполнение индивидуального задания

4.1 Описание моделей СМО

Для выполнения работы нужно создать модели СМО М/М/1 с двумя видами приоритетов (ДОБП, ДООП) для двух классов заявок.

Обозначение СМО М/М/1 в символике Кендалла значит, что:

1. интервалы времени между моментами поступления заявок в систему распределены по экспоненциальному (показательному) закону;
2. длительность обслуживания заявок в приборе распределена по экспоненциальному (показательному) закону;
3. один обслуживающий прибор в системе;
4. накопитель - неограниченной ёмкости.

Бесприоритетная дисциплина обслуживания (ДО БП) - это дисциплина обслуживания, при которой заявки выбираются на обслуживание в порядке поступления.

Дисциплина обслуживания с относительными приоритетами (ДО ОП) - это дисциплина обслуживания, при которой поступление в систему заявки с более высоким приоритетом по сравнению с обслуживаемой в приборе не приводит к прерыванию обслуживания, но при выборе заявки на обслуживание, выбирается заявка с наибольшим приоритетом.

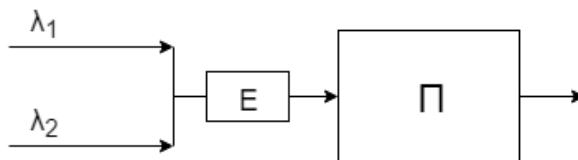


Рис. 4.1: Схема разработанной модели

4.2 Создание моделей СМО в GPSS

Для начала создадим модель СМО, показанную на Рис. 4.1, в GPSS.

Для ДО БП:

```

TRA1 VARIABLE Exponential(1,0,10)
TRA2 VARIABLE Exponential(2,0,10)
TOB1 VARIABLE Exponential(3,0,1)
TOB2 VARIABLE Exponential(4,0,1)
    GENERATE V$TRA1 ;генерирование заявок 1-го класса (загрузка = 0.1)
    ASSIGN time1,V$TOB1
    TRANSFER ,begin1
    GENERATE V$TRA2 ;генерирование заявок 2-го класса (загрузка = 0.1)
    ASSIGN time1,V$TOB2
    TRANSFER ,begin1
begin1 QUEUE Queue1
    SEIZE Uzel
    DEPART Queue1
    ADVANCE P$time1
    RELEASE Uzel
    TERMINATE 1

```

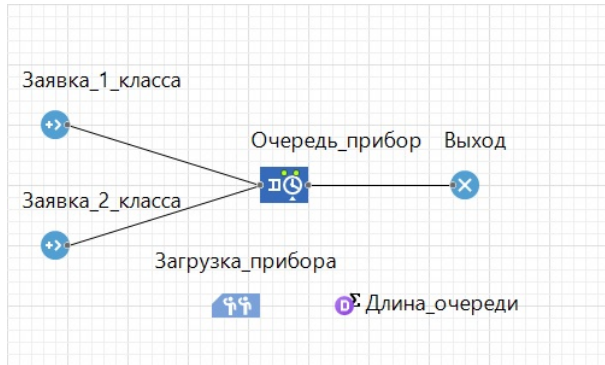
Модель с относительными приоритетами почти не отличается от вышенаписанной модели с бесприоритетной дисциплиной обслуживания, только при генерации заявок второго класса приоритет будет задаваться равным 2 вручную, а не равным 1 по умолчанию, чтобы при выборе заявки на обслуживание, выбиралась заявка второго класса(если хотя бы одна из них есть в системе), т.е. заявка с большим приоритетом:

```

TRA1 VARIABLE Exponential(1,0,10)
TRA2 VARIABLE Exponential(2,0,10)
TOB1 VARIABLE Exponential(3,0,1)
TOB2 VARIABLE Exponential(4,0,1)
    GENERATE V$TRA1
    ASSIGN time1,V$TOB1
    TRANSFER ,begin1
    GENERATE V$TRA2,,,2
    ASSIGN time1,V$TOB2
    TRANSFER ,begin1
begin1 QUEUE Queue1
    SEIZE Uzel
    DEPART Queue1
    ADVANCE P$time1
    RELEASE Uzel
    TERMINATE 1

```

4.3 Создание моделей СМО в AnyLogic



Так как нам нужно создать полностью эквивалентные модели СМО в системах GPSS World и Anylogic (т.е. модели должны рассчитывать одинаковое количество характеристик и работать без визуализации моделирования), в AnyLogic дополнительно был написан код, который считает загрузку прибора, среднюю длину очереди и среднее время ожидания, а так же перед "прогоном" модели, чтобы повысить скорость выполнения, отключим графический интерфейс, для этого будем использовать простой эксперимент, а именно напишем в свойстве нашего эксперимента следующий код:

```
getEngine().setRealTimeMode( false );  
getPresentation().getPanel().setFrameManagementAdaptive( false );  
getPresentation().getPanel().setFrameRate( 0.1 );
```

который убирает анимацию при "прогоне" модели.

Для реализации ДО ОП был написан класс заявки, хранящий переменную, обозначающую приоритет. При создании заявки, эта переменная приравнивалась 1 или 2, в зависимости от класса заявки, а на обслуживание в приборе из очереди выбиралась заявка с большим приоритетом.

```
public class Class2 extends com.anylogic.libraries.enterprise.Entity implements Serializable {  
    public double time1;  
    public int type;  
    public double start_waiting;  
    public Class2() {}  
    public Class2(double time1, int type) {  
        this.time1 = time1;  
        this.type = type;  
        this.start_waiting = start_waiting;  
    }  
    public double Start_waiting(){  
        return start_waiting;  
    }  
    public int Type(){  
        return type;  
    }  
    public double Time1(){  
        return time1;  
    }  
    @Override  
    public String toString() {  
        return
```

```

    "time1 = "+ time1 +
    "type = "+ type +
    "start_waiting ="+ start_waiting;
  }
}

```

4.4 Замеры времени моделирования в GPSS

В процессе замеров времени было проведено девять независимых экспериментов для каждой СМО: для разных соотношений загрузки классов (0.1+0.1, 0.3+0.3, 0.48+0.48), а также для разного количества пропущенных заявок (10^3 , 10^5 , 10^7).

4.5 Замеры времени моделирования в AnyLogic

Количество экспериментов для СИМ AnyLogic аналогично системе GPSS World.

Таблица 4.1: Результаты измерений СМО с ДО БП

	10^3			10^5			10^7		
	0.1+0.1	0.3+0.3	0.48+0.48	0.1+0.1	0.3+0.3	0.48+0.48	0.1+0.1	0.3+0.3	0.48+0.48
1 замер	0.4	0.4	0.4	3.4	2.8	3.3	115.5	113.7	128.5
2 замер	0.4	0.4	0.4	3.1	3.1	3.3	113	115.1	112.5
3 замер	0.4	0.4	0.5	3.9	3.1	2.8	112.6	124.9	108.6
4 замер	0.3	0.4	0.5	3.1	2.8	3.1	118.9	120.9	121.6
Ср.знач.	0.375	0.4	0.45	3.375	2.95	3.125	115	118.65	117.8

Рассчитаем доверительный интервал для времён моделирования системы с количеством заявок, равным 10^3 , 10^5 , 10^6 с доверительной вероятностью 95%. Для этого выполним прогоны имитационной модели с разными соотношениями загрузки классов (0.1+0.1, 0.3+0.3, 0.48+0.48) при неизменных параметрах. Для каждого соотношения сделаем четыре прогона и найдем среднее значение.

Таблица 4.2: Доверительные интервалы измерений СМО с ДО БП

Количество заявок	Среднее значение	С.к.о.	Дов. вероятность	Дов. интервал	Результат
10^3	0.41	0.03	95%	0.08	0.41 ± 0.08
10^5	3.15	0.17	95%	0.43	3.15 ± 0.44
10^7	117.15	1.56	95%	3.87	117.15 ± 3.88

Аналогично системе с беспriorитетной дисциплиной обслуживания выполним прогоны модели для СМО с относительными приоритетами.

Таблица 4.3: Результаты измерений СМО с ДО ОП

	10^3			10^5			10^7		
	0.1+0.1	0.3+0.3	0.48+0.48	0.1+0.1	0.3+0.3	0.48+0.48	0.1+0.1	0.3+0.3	0.48+0.48
1 замер	0.4	0.4	0.5	3.6	3.3	3.6	109.6	110.7	109.8
2 замер	0.4	0.4	0.4	2.9	2.9	3.6	108.8	112.5	110
3 замер	0.4	0.4	0.4	2.8	3.1	3.4	115.7	104	105.8
4 замер	0.4	0.4	0.4	2.4	3.6	3.2	113.2	115.5	111.6
Ср.знач.	0.4	0.4	0.425	2.925	3.225	3.45	111.825	110.675	109.3

Таблица 4.4: Доверительные интервалы измерений СМО с ДО ОП

Количество заявок	Среднее значение	С.к.о.	Дов. вероятность	Дов. интервал	Результат
10^3	0.41	0.01	95%	0.03	0.41 ± 0.03
10^5	3.20	0.22	95%	0.53	3.20 ± 0.54
10^7	110.60	1.03	95%	2.56	110.60 ± 2.57

Глава 5

Выводы по работе