

Министерство образования и науки Российской Федерации  
Санкт-Петербургский политехнический университет Петра Великого  
Институт компьютерных наук и технологий  
высшая школа киберфизических систем и управления

Работа допущена к защите  
Зав. кафедрой  
\_\_\_\_\_ Ефремов А. А.  
“ \_\_\_ ” \_\_\_\_\_ 2017г.

### **ВЫПУСКНАЯ РАБОТА БАКАЛАВРА**

Использование технологии кроссплатформенного программирования для создания ИС  
рационального использования топливных ресурсов тепловоза  
специальность 230201 “Информационные системы и технологии”

Работу выполнил  
студент гр.в43502/3 \_\_\_\_\_ / В.В.Воробьев /  
(подпись)  
Руководитель  
/ доцент, к.т.н. / \_\_\_\_\_ / С. В. Хлопин /  
(подпись)

Санкт-Петербург  
2017 г.

## РЕФЕРАТ

Работа 45 стр., 7 рис., 9 табл., 7 источников.

КРОССПЛАТФОРМЕННОЕ ПРОГРАММИРОВАНИЕ, JAVA, МЕТОД  
РАЗРЕШАЮЩИХ МНОЖИТЕЛЕЙ, МАТЕМАТИЧЕСКОЕ  
ПРОГРАММИРОВАНИЕ.

В работе рассматривается разработка информационной системы рационального использования топливных ресурсов путем оптимального распределения грузов по вагонам при помощи технологии кроссплатформенного программирования. Построена математическая модель задачи и алгоритм работы программы. Произведен анализ работы информационной системы.

Введение .....	4
1 Постановка задачи.....	7
2 Обзор базовых понятий.....	8
2.1 Классификация тепловозов .....	8
2.2 Классификация парка вагонов. ....	9
2.3 Математическое программирование. ....	15
2.4 Технология кроссплатформенного программирования .....	18
3 Построение математической модели.....	28
4 Разработка информационной системы .....	37
4.1 Алгоритм .....	37
4.2 Выполнение программы.....	39
4.3 Анализ информационной системы .....	42
Заключение .....	44
Список литературы .....	45

## **Введение**

Транспортная система России в настоящее время претерпевает качественные изменения. С одной стороны, изменяется ее структура: отдельные виды транспорта объединяются под единым управлением, в крупных транспортных узлах и административных центрах создаются независимые транспортные логистические центры, постепенно образуя единый логистический конвейер, который органично вписывается в мировую транспортную систему через существующие международные транспортные коридоры. С другой стороны, в условиях развивающихся рыночных отношений остро встают вопросы повышения эффективности работы транспорта, снижения издержек, уменьшения стоимости перевозок, соблюдения сроков доставки грузов и обеспечения их сохранности. Эти обстоятельства выдвигают принципиально новые требования к системе управления перевозками. Процессы развития экономики России, создание крупных транспортно-промышленных корпораций и международных транспортных консорциумов также требуют корректировки целей развития транспорта.

Транспортная стратегия России до 2025 г. определяет основные направления развития транспортной отрасли. Основную нагрузку в ее реализации несет железнодорожный транспорт. Железные дороги России являются особым транспортным звеном, не только связывающим промышленные центры с потребителями, но и обеспечивающим взаимодействие многих видов транспорта. Железнодорожный транспорт выполняет сегодня 85 % грузооборота и 35 % пассажирских перевозок страны. При таких масштабах сложнейшая система железнодорожного транспорта требует оптимизации. Это становится возможным только благодаря использованию новейших технологий, которые сегодня внедряются во все сферы деятельности железнодорожного транспорта.

Высокий уровень требований к рациональной работе железнодорожного транспорта определяет потребность в высоком уровне ее информатизации.

Информационные технологии сегодня становятся не просто средствами поддержки управления, а одним из важнейших элементов инфраструктуры железных дорог. Из разряда вспомогательных средств они перемещаются в класс основных технологий и являются определяющим условием совершенствования работы железнодорожного транспорта.

На железных дорогах страны разработан и успешно внедряется комплекс многоцелевых информационных технологий, позволяющий рационально использовать топливные ресурсы на базе электронного обмена данными.

Тема данной работы представляется актуальной, поскольку оптимальное использование возможностей информационной системы железных дорог в интересах всего транспортного комплекса страны позволяет:

- существенно снизить затраты на управление и связь при организации и осуществлении внутренних и международных перевозок различными видами транспорта, обеспечивает существенное повышение качества транспортных услуг;

- выполнять важнейшие социально-экономические задачи повышения производительности труда железнодорожников и качества перевозочного процесса;

- исключить потерю времени и более рационально использовать трудовые и материальные ресурсы.

Целью данной работы является создание информационной системы рационального использования топливных ресурсов тепловоза при помощи технологии кроссплатформенного программирования.

Для достижения цели необходимо решить следующие задачи:

- изучить математическое программирование;
- произвести аналитическое решение задачи при помощи оптимального метода вычисления;
- изучить технологию кроссплатформенного программирования, в частности язык Java;

- разработать алгоритм программы на основе аналитического решения задачи;

- создать информационную систему в соответствии с поставленной целью.

Объектом данного исследования является технология кроссплатформенного программирования. Предметом исследования является язык Java.

В качестве основных методов исследования использованы: анализ литературы, изучение метода линейного программирования, классификация.

В соответствии с поставленной целью и задачами выбрана структура научной работы. Работа состоит из введения, двух глав, заключения, списка использованной литературы, списка источников и двух приложений.

Во введении сформулированы тема работы и ее актуальность, определены цели и задачи, теоретическая база, методы исследования, материалы исследования и структура работы.

В первой главе формулируется постановка задачи.

Во второй главе рассматривается классификация тепловозов и вагонов, дается определение понятию математическое программирование, а также рассматривается технология кроссплатформенного программирования.

В третьей главе описывается построение математической модели.

В четвертой главе описывается алгоритм информационной системы и производится анализ ее работы.

В заключении подводятся итоги анализа, делаются основные выводы по проведенному исследованию.

## **1 Постановка задачи.**

В данной работе необходимо разработать информационную систему рационального использования топливных ресурсов тепловоза за счет оптимального распределения грузов по вагонам.

Информационная система должна соответствовать следующим требованиям:

- позволять вводить данные по количеству вагонов и груза;
- делать возможным для пользователя изменять характеристики вагонов и груза;
- производить вычисление оптимального количества вагонов;
- выводить результат вычисления в виде таблицы;
- позволять пользователю распечатать результат или сохранить в виде файла;
- работать на различных операционных системах (в частности на Windows и Linux);

Прежде чем приступить к построению математической модели необходимо рассмотреть классификацию тепловозов, парка вагонов и дать понятие методам математического программирования.

## **2 Обзор базовых понятий**

Ввиду того, что в настоящее время на предприятии ОАО «РЖД» в локомотивном депо отсутствуют методики рационального использования топливных ресурсов тепловоза, ниже будет представлена математическая модель решения данной задачи путем оптимального распределения грузов по вагонам.

Прежде чем приступить к построению математической модели необходимо рассмотреть базовые понятия объектов железнодорожного транспорта, методы математического программирования и технологию кроссплатформенного программирования.

### **2.1 Классификация тепловозов**

По роду службы тепловозы классифицируются на поездные, маневровые и промышленные. В свою очередь среди поездных, или магистральных, выделяют грузовые, пассажирские и грузопассажирские. Назначение тепловоза определяется его техническими характеристиками — так, для грузовых тепловозов важна в первую очередь значительная сила тяги, тогда как пассажирским нужна высокая скорость. Маневровые и промышленные локомотивы обычно используются для передвижения вагонов в пределах станции или на подъездных путях предприятий с малыми радиусами кривых. Именно поэтому большинство таких локомотивов — тепловозы, так как для работы на любых, в том числе неэлектрифицированных вспомогательных путях, важна автономность энергетической установки. Основные параметры тепловозов приведены в таблице 1.



Таблица 1

Серия	Назначение	Сцепная масса, т	Конструкционная скорость км/ч	Мощность по дизелю, кВт
ТЭЗ	Грузовой	2 х 126	100	2 х 1470
2ТЭ10В	Грузовой	2 х 129	100	2 х 2210
ТЭП69	Пассажирский	129	160	2210
2М62	Грузовой	2 х 120	100	2 х 1470
ТЭМ2	Маневровый	120	100	880
2ТЭ116	Грузовой	2 х 138	100	2 х 2210
ТЭП70	Пассажирский	129	160	2940
ТЭМ7	Маневровый	180	100	1470

## 2.2 Классификация парка вагонов.

В состав вагонного парка входят пассажирские и грузовые вагоны.

В зависимости от технических характеристик вагоны классифицируют следующим образом: по числу осей (четырёх-, шести-, восьми- и многоосные); по виду материала и технологии изготовления кузова (цельнометаллические, с деревянной или металлической обшивкой, с кузовом из легких сплавов); по грузоподъемности, массе тары вагона, нагрузке на 1 погонный метр пути, габариту подвижного состава и другим показателям.

В состав парка грузовых вагонов входят крытые вагоны, платформы, полувагоны, цистерны, изотермические вагоны и вагоны специального назначения.

Крытые вагоны предназначены для перевозки разнообразных грузов, обеспечения их сохранности и защиты от воздействия атмосферы. Эти вагоны, оснащенные соответствующим оборудованием, могут быть использованы и для массовой перевозки людей. Кузов крытого вагона имеет в каждой из боковых стен задвижные двери и по два люка с металлическими крышками. Люки

служат для освещения, вентиляции и загрузки вагонов сыпучими грузами. Крытые вагоны, выпускаемые в настоящее время, имеют металлический кузов и расширенный дверной проем. Грузоподъемность вагона 68 т, вместимость кузова 140 м<sup>3</sup>.

На платформах перевозят длинномерные, громоздкие и тяжеловесные грузы. Платформы оборудуют невысокими откидными металлическими бортами и приспособлениями для установки стоек, необходимых при перевозке бревен, столбов, досок и т.п. Грузоподъемность современных платформ составляет 70...72 т. Для перевозки крупнотоннажных контейнеров массой брутто 10, 20 и 30 т выпускают специальные четырехосные платформы, снабженные фитингами – устройствами для установки и крепления контейнеров.

Полувагоны – наиболее распространенный тип вагонов грузового парка. Они служат в основном для перевозки навалочных сыпучих грузов, таких, как уголь, руда, кокс, щебень, гравий и др. В полу кузова, вдоль боковых стен, предусмотрены разгрузочные люки, через которые сыпучий груз самотеком разгружается по обе стороны полувагона. Погрузку в полувагон длинномерных грузов и самоходного транспорта осуществляют через двери.

На железных дорогах применяют четырех- и восьмиосные полувагоны, у которых боковые стены и торцевые двери кузова имеют металлическую обшивку. Выпускают также полувагоны с глухим кузовом, без разгрузочных люков; их разгружают на вагоноопрокидывателях.

Разновидностью полувагонов являются так называемые вагоны-хопперы для перевозки сыпучих и пылевидных грузов (щебень, гравий, песок, цемент, зерно и др.) грузоподъемностью 50 т. Хопперы имеют высокие боковые стены. Для перевозки грузов, которые необходимо защитить от атмосферных осадков, используют полувагоны с крышей. Их торцевые стены наклонены к середине вагона, где расположены разгрузочные люки.

На внутренних путях крупных металлургических заводов руду и строительные сыпучие материалы перевозят преимущественно полувагонами-

самосвалами, называемыми думпкарами. Это четырехосные полувагоны грузоподъемностью 60 т и более с кузовом прямоугольной формы, снабженные пневматическим устройством для разгрузки, при выполнении которой кузов наклоняется и одновременно открывается борт с соответствующей стороны.

Жидкие грузы (нефть, керосин, бензин, масло, кислоты и т.п.) перевозят в цистернах. Цистерна представляет собой специальный металлический сварной резервуар (котел) цилиндрической формы, имеющий в верхней части люки для наливания груза, очистки и ремонта. Разнообразие грузов обуславливает существенные различия в конструкции цистерн.

В зависимости от вида перевозимых грузов цистерны могут быть разделены на две группы:

- общего назначения – для перевозки нефтепродуктов широкой номенклатуры;
- специальные – для перевозки отдельных видов грузов.

Цистерны общего назначения подразделяют на используемые для перевозки светлых (бензин, лигроин и т.п.) и темных (нефть, масла и т.п.) нефтепродуктов. Внутренняя поверхность цистерн, в которых перевозят кислоты, покрыта защитным слоем (резина, свинец), предохраняющим металл от разрушающего действия кислот. В этих же целях котлы цистерн изготавливают из кислотоупорных металлов – коррозионно-стойкой стали, алюминия. Цистерны для перевозки молока выполняют из аналогичной стали, покрытой снаружи теплоизолирующим слоем.

Вязкие нефтепродукты перевозят в цистернах, оборудованных паровой рубашкой, что значительно упрощает и ускоряет слив предварительно разогретых грузов. Четырехосные цистерны имеют котел вместимостью  $72\text{ м}^3$ . Применяются и восьмиосные цистерны с котлом вместимостью  $134\text{ м}^3$ .

Изотермические вагоны используют в летнее время для перевозки скоропортящихся грузов (мясо, рыба и др.), а зимой – грузов, теряющих свои качества при замерзании (овощи, фрукты, молоко и др.). Для поддержания в

вагонах необходимой температуры их оборудуют приборами охлаждения и отопления, а кузова снабжают тепловой изоляцией.

Изотермические вагоны соединяют в рефрижераторные секции по пять единиц. При этом в одном вагоне размещаются обслуживающая бригада механиков, дизель-электростанция и холодильное оборудование.

Для перевозки скоропортящихся грузов применяют также автономные рефрижераторные вагоны, оборудованные холодильными агрегатами и дизель-генераторными установками с автоматическим (без обслуживающего персонала) управлением.

Помимо универсальных изотермических вагонов, используемых для перевозки скоропортящихся грузов, находятся в эксплуатации и специализированные вагоны для транспортирования живой рыбы, молочных и других продуктов.

Вагоны специального назначения предназначены для грузов, требующих особых условий перевозки. Например, транспортерами перевозят громоздкие и тяжеловесные машины и оборудование. Транспортеры – это многоосные платформы (12, 16, 20 и более осей) грузоподъемностью 130, 180, 230 и 300 т. К специальным относятся также вагоны для перевозки скота, живой рыбы, битума, легковых автомобилей и вагоны, предназначенные для технических и бытовых нужд железных дорог: вагоны-мастерские, вагоны восстановительных и пожарных поездов. Состав оборудования этих вагонов определяется их назначением.

Для перевозки различных грузов, в том числе штучных изделий, домашних вещей и др., используют деревянные или металлические контейнеры с массой брутто 3, 5, 20 т и более. При перевозке на платформах или в полувагонах контейнеры закрепляют соответствующими приспособлениями. Чтобы избежать перегрузки из вагонов в автомашины, применяют специальные контейнеры большой грузоподъемности, приспособленные для подкатки под них автомобильных шасси. Такие контейнеры называют контрейлерами.

В последние годы начат выпуск вагонов нового поколения, к которым можно отнести вагоны с раздвижными колесными парами, крытые вагоны с открывающейся или сдвигающейся крышей и со встроенными ленточными конвейерами для полной механизации выгрузки картофеля, овощей и фруктов, платформы со специальными стационарными приспособлениями для перевозки лесоматериалов и металлопроката, специальные вагоны с кузовами-амфибиями для транспортирования навалочных грузов в смешанном железнодорожно-водном сообщении.

Технико-экономические показатели вагонов.

Основными показателями, необходимыми для технико-экономической оценки конструкции и эксплуатационных особенностей вагонов, являются число осей, грузоподъемность, тара, коэффициент тары, удельный объем кузова, удельная площадь пола, давление колесной пары на рельсы, давление вагона, приходящееся на 1 погонный метр пути.

С числом осей связана грузоподъемность вагона – наибольшая масса груза, которая может быть перевезена, исходя из прочности конструкции вагона. Достоинства вагонов большой грузоподъемности таковы:

- меньшее удельное сопротивление движению, за счет чего сокращается расход электроэнергии и топлива, потребляемых локомотивами;
- большая погонная нагрузка, т.е. масса поездов возрастает при неизменной длине станционных путей;
- снижение металлоемкости конструкции на единицу грузоподъемности на 10...15%;
- сокращение расходов на ремонт и содержание вагонов на 10...20%;
- снижение затрат на маневровую работу, взвешивание вагонов и оформление перевозочной документации.

Сумма грузоподъемности вагона (масса нетто) и его тары составляет массу вагона брутто. Уменьшение тары вагонов, представляющее собой одну из основных задач вагоностроения, обеспечивает увеличение грузоподъемности грузовых вагонов и, следовательно, повышение провозной способности

железных дорог, экономию металла, необходимого для постройки вагонов, электроэнергии и топлива, расходуемых локомотивами при перевозке, а также снижение себестоимости перевозок.

Наиболее важным показателем, характеризующим технико-экономическую эффективность вагона, является коэффициент тары

$$K_T = T/P,$$

где  $T$  – тара вагона;  $P$  – его грузоподъемность.

Этот коэффициент показывает, какая часть массы вагона приходится на каждую тонну его грузоподъемности. Чем меньше коэффициент тары, тем экономичнее вагон. Для пассажирских вагонов коэффициент тары определяется как отношение тары вагона к числу мест.

Показателем вместимости вагона служит удельный объем кузова  $v_y$ , а у платформ – удельная площадь пола  $f_y$ :

$$v_y = V/P; f_y = F/P,$$

где  $V$  – вместимость кузова вагона;  $F$  – площадь пола платформы.

Воздействие вагонов на верхнее строение пути и искусственные сооружения (мосты, путепроводы) характеризуется силой, с которой колесная пара действует на рельсы, т.е. осевым давлением, и давлением на 1 погонный метр пути.

Допустимая нагрузка зависит от прочности железнодорожного пути и мощности элементов его верхнего строения (иначе говоря от типа рельсов, числа шпал на 1 км пути, вида балласта). Исходя из этого, сила, с которой колесная пара действует на рельс, для грузовых вагонов на наших дорогах ограничена 228 кН. Допустимая нагрузка определяется прочностью искусственных сооружений и для основных типов вагонов составляет 88 кН на 1 погонный метр пути.

Возрастание допустимой нагрузки позволяет при той же длине станционных путей увеличить массу поездов и, следовательно, повысить

провозную способность железной дороги. Таким образом, допустимая нагрузка определяет и грузоподъемность вагонов.

Таблица 2

Тип вагона	Число осей	Тара, т	Грузо-подъемность, т	Коэффициент тары	Вместимость кузова, м <sup>3</sup>	Длина вагона, м
Крытый	4	22,9	64	0,35	90...120	14,73
Полувагон	4	22,1	65	0,34	72,5	13,92
	8	45,5	125	0,35	137,5	20,24
Платформа	4	21,0	62...66	0,32	36,8	14,62
Цистерна	4	23,1	62	0,39	72,7	12,02
	8	48,8	120	0,41	137,2	21,12
Транспортер	20	142,0	300	0,47	-	45,0

При проектировании вагонов устанавливают исходя из заданного габарита подвижного состава объем кузова, а для платформ – площадь пола и затем по этим данным находят внутренние размеры вагонов.

При выборе длины вагона учитывают вынос его кузова в кривых участках пути и условия размещения в вагонах грузов и контейнеров.

Основные характеристики и технико-экономические показатели наиболее распространенных конструкций грузовых вагонов приведены в табл. 2.

### **2.3 Математическое программирование.**

Математическое программирование – это математическая дисциплина, в которой разрабатываются методы отыскания экстремальных значений целевой функции среди множества ее возможных значений, определяемых ограничениями.

Наличие ограничений делает задачи математического программирования принципиально отличными от классических задач математического анализа по

отысканию экстремальных значений функции. Методы математического анализа для поиска экстремума функции в задачах математического программирования оказываются непригодными.

Для решения задач математического программирования разработаны и разрабатываются специальные методы и теории. Так как при решении этих задач приходится выполнять значительный объем вычислений, то при сравнительной оценке методов большое значение придается эффективности и удобству их реализации на ЭВМ.

Математическое программирование можно рассматривать как совокупность самостоятельных разделов, занимающихся изучением и разработкой методов решения определенных классов задач.

В зависимости от свойств целевой функции и функции ограничений все задачи математического программирования делятся на два основных класса:

- задачи линейного программирования,
- задачи нелинейного программирования.

Если целевая функция и функции ограничений – линейные функции, то соответствующая задача поиска экстремума является задачей линейного программирования. Если хотя бы одна из указанных функций нелинейна, то соответствующая задача поиска экстремума является задачей нелинейного программирования.

Линейное программирование (ЛП) – один из первых и наиболее подробно изученных разделов математического программирования. Именно линейное программирование явилось тем разделом, с которого и начала развиваться сама дисциплина "математическое программирование". Термин "программирование" в названии дисциплины ничего общего с термином "программирование (т.е. составление программы) для ЭВМ" не имеет, т.к. дисциплина "линейное программирование" возникла еще до того времени, когда ЭВМ стали широко применяться для решения математических, инженерных, экономических и др. задач.



Термин "линейное программирование" возник в результате неточного перевода английского "linear programming". Одно из значений слова "programming" - составление планов, планирование. Следовательно, правильным переводом английского "linear programming" было бы не "линейное программирование", а "линейное планирование", что более точно отражает содержание дисциплины. Однако, термины линейное программирование, нелинейное программирование, математическое программирование и т.д. в нашей литературе стали общепринятыми и поэтому будут сохранены.

Итак, линейное программирование возникло после второй мировой войны и стало быстро развиваться, привлекая внимание математиков, экономистов и инженеров благодаря возможности широкого практического применения, а также математической стройности.

Можно сказать, что линейное программирование применимо для решения математических моделей тех процессов и систем, в основу которых может быть положена гипотеза линейного представления реального мира.

Линейное программирование применяется при решении экономических задач, в таких задачах как управление и планирование производства; в задачах определения оптимального размещения оборудования на морских судах, в цехах; в задачах определения оптимального плана перевозок груза (транспортная задача); в задачах оптимального распределения кадров и т.д.

Задача линейного программирования (ЛП), как уже ясно из сказанного выше, состоит в нахождении минимума (или максимума) линейной функции при линейных ограничениях. Общий вид задачи линейного программирования:

$$f(x) = \sum_{j=1}^n c_j x_j = c_1 x_1 + c_2 x_2 + \dots + c_n x_n$$

Ограничения:

$$g_1(x_1, \dots, x_n) \leq b_1;$$

$$g_2(x_1, \dots, x_n) \leq b_2;$$

$$g_m(x_1, \dots, x_n) \leq b_m;$$

где  $f(x)$  - целевая функция, или критерий эффективности (например, прибыль от производства каких-либо видов продукции, стоимость перевозок и т.п.);  $X=\{x_1, \dots, x_n\}$  - варьируемые параметры;  $g_1(x), \dots, g_m(x)$  - функции, которые задают ограничения на имеющиеся ресурсы.

## **2.4 Технология кроссплатформенного программирования**

Кроссплатформенность — способность программного обеспечения работать более чем на одной аппаратной платформе и (или) операционной системе.

Типичным примером является программное обеспечение, предназначенное для работы в операционных системах Linux и Windows одновременно. Одним из способов достижения кроссплатформенности является использование кроссплатформенных языков программирования, поддерживающих разработку на них программ для выполнения под различные платформы. Другим способом является использование кроссплатформенных сред исполнения, к особенностям которых является исполнение средой не непосредственно специфических для определенного процессора машинных команд, а промежуточного кода. При этом среда выполнения играет роль «прослойки», осуществляющей трансляцию промежуточного кода прикладной программы в машинные коды и их выполнение. Кроме этих способов достижения кроссплатформенности, также существует возможность выполнения программы, не предназначенной для выполнения под определенной платформой посредством использования эмулятора базовой платформы, требуемой для исполнения программы. При этом подходе программа исполняется в среде эмулятора. Обычно исполнение программы в среде эмулятора приводит к снижению производительности по сравнению с аналогичными программами, для которых платформа является базовой, т.к. значительная часть ресурсов системы расходуется на выполнение функций эмуляции. В то же время, этот способ позволяет достичь кроссплатформенного

исполнения программ, которые были разработаны без учета требований к кроссплатформенности.

Для разработки информационной системы будет использоваться язык Java. Рассмотрим подробнее этот язык.

Java – один из самых важных и широко применяемых языков программирования в мире на протяжении многих лет. В отличие от некоторых других языков программирования, влияние Java не только не уменьшилось со временем, а, наоборот, возросло. То, что язык Java за короткое время стал одним из самых популярных коммерческих объектно-ориентированных языков программирования, помогает распространению технологии Java и подтверждается тем, что в настоящее время этот язык использует около 1 млн. разработчиков (80% из них создает кросс-платформенные приложения), продукты для Java выпускают сотни фирм, а число копий JDK, загруженных с сайта фирмы Sun, превысило 2.5 млн.

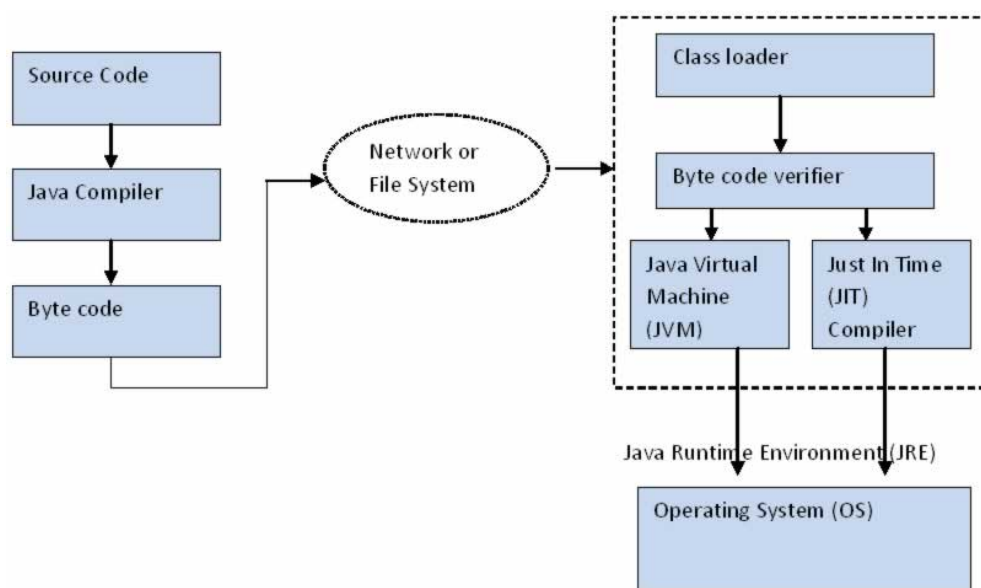


Рисунок 1 – архитектура Java.

Язык Java тесно связан с языком C++, который, в свою очередь, является прямым наследником языка C. Многие особенности Java унаследованы от обоих языков. От C язык Java унаследовал свой синтаксис, а многие его объектно-ориентированные свойства были перенесены из C++. Собственно

говоря, ряд определяющих характеристик языка Java был перенесен – или разработан в ответ на возникшие потребности – из его предшественников. Более того, создание Java своими корнями уходит глубоко в процесс усовершенствования и адаптации, который происходит в языках программирования на протяжении нескольких последних десятилетий. Каждое новшество в архитектуре языка, как правило, было обусловлено необходимостью найти решение той или иной основной проблемы, которую не могли разрешить существовавшие до этого языки. И Java не является исключением из этого правила.

Как упоминалось ранее, язык Java наследует многие из своих характеристик от языков C и C++. Это сделано намеренно. Разработчики Java знали, что использование знакомого синтаксиса C и повторение объектно-ориентированных свойств C++ должно было сделать их язык привлекательным для миллионов, имеющих опыт программирования на C/C++. Помимо внешнего сходства, в Java используется ряд других характерных особенностей, которые способствовали успеху языков C и C++. Во-первых, язык Java был разработан, проверен и усовершенствован людьми, имевшими солидный практический опыт программирования. Этот язык разработан с учетом потребностей и опыта его создателей. Таким образом, Java – это язык для программистов. Во-вторых, Java целостен и логически непротиворечив. В-третьих, Java предоставляет программисту полный контроль над программой, если не учитывать ограничения, накладываемые средой Интернета. Если программирование выполняется грамотно, это непосредственно отражается на самих программах. В равной степени справедливо и обратное. Иначе говоря, Java – этот язык не для обучающихся программированию, а для тех, кто занимается им профессионально.

Из-за сходства характеристик языков Java и C некоторые склонны считать Java просто «версией языка C++ для Интернета». Но это серьезное заблуждение. Языку Java присущи значительные практические и концептуальные отличия. Язык C++ действительно оказал влияние на

характеристики Java, но Java не является усовершенствованной версией C++. Например, Java несовместим с языком C++. Разумеется, у него немало сходств с языком C++, и в исходном коде программы на Java программирующий на C++ будет чувствовать себя почти как дома. Вместе с тем язык Java предназначен не для замены C++, а для решения одних задач, тогда как язык C++ – для решения других. Оба этих языка будут еще долго сосуществовать.

Основным фактором, обусловившим изобретение Java, стала потребность в обеспечении переносимости и безопасности, но свою роль в формировании окончательной версии языка сыграли и другие факторы. Группа разработчиков обобщила основные понятия Java и составила следующий перечень его особенностей:

- простота;
- безопасность;
- переносимость;
- объектная ориентированность;
- надежность;
- многопоточность;
- архитектурная нейтральность;
- интерпретируемость;
- высокая производительность;
- распределенность;
- динамичность.

К слову о простоте. Язык Java был задуман как простой в изучении и эффективный в употреблении профессиональными программистами. Овладеть языком Java тем, у кого имеется некоторый опыт программирования, не составит особого труда. Язык Java наследует синтаксис C/C++ и многие объектно-ориентированные свойства C++, поэтому для большинства программистов изучение Java не составит больших трудностей.

Если говорить о переносимости программ и обеспечении безопасности, то основная особенность Java состоит в том, что компилятор Java выдает не

исполняемый код, а так называемый байт-код – в высшей степени оптимизированный набор инструкций, предназначенных для выполнения в исполняющей системе Java, называемой виртуальной машиной Java (Java Virtual Machine – JVM). Собственно говоря, первоначальная версия виртуальной машины JVM разрабатывалась в качестве интерпретатора байт-кода. Это может вызвать недоумение, поскольку для обеспечения максимальной производительности компиляторы многих современных языков программирования призваны создавать исполняемый код. Но то, что программа на Java интерпретируется виртуальной машиной JVM, как раз помогает решить основные проблемы разработки программ.

Трансляция программы Java в байт-код значительно упрощает ее выполнение в разнотипных средах, поскольку на каждой платформе необходимо реализовать только виртуальную машину JVM. Если в отдельной системе имеется исполняющий пакет, в ней можно выполнять любую программу на Java. Следует, однако, иметь в виду, что все виртуальные машины JVM на разных платформах, несмотря на некоторые отличия и особенности их реализации, способны правильно интерпретировать один и тот же байт-код. Если бы программа на Java компилировалась в машинозависимый код, то для каждого типа процессоров должны были бы существовать отдельные версии одной и той же программы. Ясно, что такое решение неприемлемо. Таким образом, организация выполнения байт-кода виртуальной машиной JVM – простейший способ создания по-настоящему переносимых программ.

Тот факт, что программа на Java выполняется виртуальной машиной JVM, способствует также повышению ее безопасности. Виртуальная машина JVM управляет выполнением программы, поэтому она может изолировать программу и воспрепятствовать возникновению побочных эффектов от ее выполнения за пределами данной системы. Как станет ясно в дальнейшем, ряд ограничений, существующих в языке Java, также способствует повышению безопасности.

В общем, когда программа компилируется в промежуточную форму, а затем интерпретируется виртуальной машиной JVM, она выполняется медленнее, чем если бы она была скомпилирована в исполняемый код. Но в Java это отличие в производительности не слишком заметно. Байт-код существенно оптимизирован, и поэтому его применение позволяет виртуальной машине JVM выполнять значительно быстрее, чем следовало ожидать.

Язык Java был задуман как интерпретируемый, но ничто не препятствует ему оперативно выполнять компиляцию байт-кода в машинозависимый код для повышения производительности. Поэтому вскоре после выпуска Java появилась технология HotSpot, которая представляет динамический компилятор (или так называемый JIT-компилятор) байт-кода. Если динамический компилятор входит в состав виртуальной машины JVM, то избранные фрагменты байт-кода компилируются в исполняемый код по частям, в реальном времени и по требованию. Важно понимать, что одновременная компиляция всей программы Java в исполняемый код нецелесообразна, поскольку Java производит различные проверки, которые могут быть сделаны только во время выполнения. Вместо этого динамический компилятор компилирует код во время выполнения по мере надобности. Более того, компилируются не все фрагменты байт-кода, а только те, которым компиляция принесет выгоду, а остальной код просто интерпретируется. Тем не менее принцип динамической компиляции обеспечивает значительное повышение производительности. Даже при динамической компиляции байт-кода характеристики переносимости и безопасности сохраняются, поскольку виртуальная машина JVM по-прежнему отвечает за целостность исполняющей среды.

Касательно объектной ориентированности, хотя предшественники языка Java и оказали влияние на его архитектуру и синтаксис, при его проектировании не ставилась задача совместимости по исходному коду с каким-нибудь другим языком. Это позволило группе разработчиков создавать Java, по существу, с чистого листа. Одним из его следствий этого явился четкий, практичный, прагматичный подход к объектам. Помимо того, что язык

Java позаимствовал свойства многих удачных объектно-программных сред, разработанных на протяжении нескольких последних десятилетий, в нем удалось достичь золотой середины между строгим соблюдением принципа «все элементы программы являются объектами» и более прагматичного принципа «прочь с дороги». Объектная модель Java проста и легко расширяема. В то же время такие элементарные типы данных, как целочисленные, сохраняются в виде высокопроизводительных компонентов, не являющихся объектами.

Если говорить о надежности, то будет уместно упомянуть, что многоплатформенная среда предъявляет к программам повышенные требования, поскольку они должны надежно выполняться в разнотипных системах. Поэтому способность создавать надежные программы была одним из главных приоритетов при разработке Java. Для обеспечения надежности в Java накладывается ряд ограничений в нескольких наиболее важных областях, что вынуждает программистов выявлять ошибки на ранних этапах разработки программы. В то же время Java избавляет от необходимости беспокоиться по поводу многих наиболее часто встречающихся ошибок программирования. А поскольку Java – строго типизированный язык, то проверка кода выполняется во время компиляции. Но проверка кода делается и во время выполнения. В результате многие трудно обнаруживаемые программные ошибки, которые часто приводят к возникновению с трудом воспроизводимых ситуаций во время выполнения, попросту невозможны в программе на Java. Предсказуемость кода в разных ситуациях – одна из основных особенностей Java.

Чтобы понять, каким образом достигается надежность программ на Java, рассмотрим две основные причины программных сбоев: ошибки управления памятью и неправильная обработка исключений (т.е. ошибки при выполнении). В традиционных средах создания программ управление памятью – сложная и трудоемкая задача. Например, в среде C/C++ программист должен вручную резервировать и освобождать всю динамически распределяемую память. Иногда это ведет к возникновению трудностей, поскольку программисты забывают освободить ранее зарезервированную память или, что еще хуже,



пытаются освободить область памяти, все еще используемую другой частью кода. Java полностью исключает такие ситуации, автоматически управляя резервированием и освобождением памяти. (Освобождение оперативной памяти полностью выполняется автоматически, поскольку Java предоставляет средства сборки неиспользуемых объектов в «мусор».) В традиционных средах условия для исключений часто возникают в таких ситуациях, как деление на ноль или отсутствие искомого файла, а управление ими должно осуществляться с помощью громоздких и трудных для понимания конструкций. Java облегчает выполнение этой задачи, предлагая объектно-ориентированный механизм обработки исключений. В грамотно написанной программе на Java все ошибки при выполнении могут (и должны) обрабатываться самой программой.

Что же касается многопоточности, то язык Java был разработан в ответ на потребность создавать интерактивные сетевые программы. Для этой цели в Java поддерживается написание многопоточных программ, способных одновременно выполнять многие действия. Исполняющая система Java содержит изящное, но вместе с тем сложное решение задачи синхронизации многих процессов, которое позволяет строить устойчиво работающие интерактивные системы. Простой подход к организации многопоточной обработки, реализованный в Java, позволяет программистам сосредоточивать основное внимание на конкретном поведении программы, а не на создании многозадачной подсистемы.

Говоря об архитектурной нейтральности следует отметить, что основной задачей, которую ставили перед собой разработчики Java, было обеспечение долговечности и переносимости кода. Одной из главных трудностей, стоявших перед разработчиками, когда они создавали Java, было отсутствие всяких гарантий, что код, написанный сегодня, будет успешно выполняться завтра – даже на одном и том же компьютере. Операционные системы и процессоры постоянно совершенствуются, и любые изменения в основных системных ресурсах могут стать причиной неработоспособности программ. Пытаясь каким-то образом изменить это положение, разработчики приняли ряд жестких

решений в самом языке и виртуальной машине Java. Они поставили перед собой следующую цель: «написано однажды, выполняется везде, в любое время и всегда». И эта цель была в значительной степени достигнута.

Таблица 3

Особенность	Язык		
	Java	C++	C#
Кроссплатформенность	+	–	–
Безопасность	+	–	+
Многопоточность	+	+	+
Простота	+	–	+
Надежность	+	–	–
Интерпретируемость	+	–	+
Открытый код	+	–	–

К слову об интерпретируемости и высокой производительности. Как упоминалось ранее, выполняя компиляцию программ в промежуточное представление, называемое байт-кодом, Java позволяет создавать межплатформенные программы. Такой код может выполняться в любой системе, на которой реализована виртуальная машина JVM. С первых же попыток разработать межплатформенные решения удалось достичь поставленной цели, хотя и за счет снижения производительности. Как пояснялось ранее, байт-код Java был тщательно разработан таким образом, чтобы его можно было с высокой эффективностью преобразовать непосредственно в машинозависимый код на конкретной платформе с помощью динамического компилятора. Исполняющие системы Java, обеспечивающие такую возможность, сохраняют все преимущества кода, не зависящего от конкретной платформы.

В заключении о динамичности. Программы на Java содержат значительный объем данных динамического типа, используемых для проверки

полномочий и разрешения доступа к объектам во время выполнения. Это позволяет безопасно и рационально выполнять динамическое связывание кода. Данное обстоятельство исключительно важно для устойчивости среды Java, где небольшие фрагменты байт-кода могут динамически обновляться в действующей системе.

В табл. 3 приведено сравнение языка Java с языками C++ и C#. По сравнению с двумя последними языками Java имеет определенные преимущества. В связи с этим в данной работе в качестве технологии кроссплатформенного программирования был выбран этот язык.

### 3 Построение математической модели.

В термин «транспортной задачи» входит широкий круг поставленных задач с различными математическими моделями. Классический вид данных задач – это выбор наиболее выгодного с экономической точки зрения плана перевозок различных продуктов из пунктов их производства в пункты потребления. Решать задачи такого рода проще всего, применяя линейное программирование. Оно является одной из областей математики, где разработаны теории и численные методы решения многомерных задач, включающих в себя различные ограничения.

Большое количество вариантов перевозок, которые возможно совершить, создает затруднительную ситуацию в выборе наиболее экономичного плана эмпирическим или же экспертным путем. Применение линейного программирования в решении транспортной задачи дает положительный эффект с экономической точки зрения.

Обозначим количество вагонов определенного типа через  $A_i$  ( $i = 1, 2, \dots, m$ ), где  $m$  - тип вагона (крытый, платформа, цистерна, и т.п.); объем погрузки каждого груза в тоннах  $B_j$  ( $j = 1, 2, \dots, n$ ), где  $n$  – тип груза (уголь, глина, дерево, и т. п.); техническую норму загрузки через  $p_{ij}$  вагонов типа  $i$  грузом рода  $j$ . Если загрузить данный груз в вагоны данного типа невозможно,  $p_{ij} = 0$ . Обозначив количество вагонов типа  $i$ , подаваемых под погрузку груза  $j$ , через  $x_{ij}$ , выразим условия задачи следующим образом. Требуется минимизировать целевую функцию

$$\sum_{i=1}^m \sum_{j=1}^n x_{ij} = \min$$

при следующих ограничениях:

$$\left. \begin{aligned} \sum_{j=1}^n x_{ij} &\leq A_i, & (i = 1, 2, \dots, m); \\ \sum_{i=1}^m p_{ij} x_{ij} &= B_j, & (j = 1, 2, \dots, n); \\ x_{ij} &\geq 0, & (i = 1, 2, \dots, m; j = 1, 2, \dots, n). \end{aligned} \right\}$$

Целевая функция выражает общее количество порожних вагонов, необходимых для погрузки всех грузов. Первое ограничение системы означает, что превысить ресурсы порожних вагонов нельзя; второе говорит о том, что необходимо выполнить задание (заявки) на погрузку каждого груза; третье отражает тот факт, что отрицательные значения  $x_{ij}$  в подобных задачах не имеют смысла.

В поставленной задаче в качестве ресурсов рассматриваются порожние вагоны и минимизируется их общее количество, необходимое для погрузки всех грузов, что равносильно максимизации средней статической нагрузки вагона.

Распределительную задачу можно решить универсальным симплекс-методом, однако существуют специальные методы, которые значительно проще. Один из наиболее эффективных – метод разрешающих множителей – разработан акад. Л. В. Канторовичем. Его особенность – простота и возможность получения целочисленного результата.

Вначале составляется план, обеспечивающий минимум целевой функции, но не отвечающий ограничениям. Все исходные данные заносятся в таблицу-матрицу. Каждая ее строка, кроме последней, соответствует одному типу подвижного состава, каждый столбец, кроме двух последних, – одному роду груза. В левом верхнем углу каждой клетки проставляется техническая норма загрузки вагона данного типа соответствующим грузом  $p_{ij}$ . Предпоследний столбец предназначен для общего количества вагонов каждого типа, назначаемых под погрузку на каждом этапе решения задачи, а в последнем –

указывают величину избытка или их недостатка. В последнюю строку в ходе решения записывают разрешающие множители  $\lambda$ .

Затем отыскивается допустимый оптимальный план. Для этого в каждом столбце отыскивается клетка с максимальной технической нормой погрузки, в нее заносится число имеющихся тонн груза и число вагонов, необходимых для перевозки этого груза (погрузка). По условию допускается перегруз, равный 2 тоннам. Для проверки перегруза необходимо поделить вес на целое число полученных вагонов. Разница между грузоподъемностью вагона и полученным весом в результате деления будет перегрузкой. Назначение вагонов показываем в матрице дробью, в числителе которой число вагонов, а в знаменателе – количество загружаемого в них груза;

Подразделяем строки матрицы на недостаточные и избыточные, для чего по составленному плану подсчитываем общую потребность вагонов в каждой строке и сравниваем ее с наличием. Если вагонов не хватает для покрытия потребности, то в последнем столбце проставляем величину недостатка со знаком минус, и строка считается недостаточной. При избытке вагонов его величина записывается со знаком плюс, и строка считается избыточной. Строку, где наличие вагонов и потребность в них равны, считаем недостаточной, если найдутся два элемента, принадлежащие этой и другой какой-либо недостаточной строке и стоящие в одном столбце, в которых технические нормы загрузки  $p_{ij}$  одинаковы, а в элементе недостаточной строки имеются вагоны. В противном случае строка считается избыточной. Если все строки окажутся избыточными или недостаточными, то начальный план будет оптимальным. При наличии хотя бы одной недостаточной строки (или хотя бы одной избыточной, когда остальные недостаточные) необходимо дальнейшее улучшение плана;

Для корректировки неоптимального плана для каждого столбца находится разрешающий множитель  $\lambda_j$  по формуле:  $\lambda_j = \frac{P_j^{\text{зан}(-)}}{P_j^{\text{max}}}$ , где  $P_j^{\text{зан}(-)}$  –

техническая норма погрузки в занятой клетке недостающей строки;  $P_j^{max}$  – максимальная техническая норма погрузки в избыточных строках. Разрешающие множители всех столбцов сравниваются между собой и к дальнейшим расчетам принимается минимальный разрешающий множитель.

Все технические нормы погрузки в избыточных строках (+) умножаются на выбранный разрешающий множитель, или в недостаточных строках (-) делятся на него. Через клетки с выровненными техническими нормами погрузки осуществляется корректировка распределения вагонов. Корректировка осуществляется с учетом истинных значений технических норм погрузки.

Итерацию повторяют до тех пор, пока не получат матрицу лишь с одними избыточными или с одними недостаточными строками. Это и будет оптимальный вариант распределения порожних вагонов под погрузку. Возможен также оптимальный вариант, когда все избытки и недостатки равны нулю, т.е. для погрузки всех грузов необходимо ровно столько вагонов, сколько их имеется. Наличие в матрице одних недостаточных строк говорит о том, что парка вагонов не хватает для погрузки всех заявленных грузов, причем найденный вариант обеспечивает погрузку максимального их количества. При наличии в матрице одних избыточных строк часть вагонного парка излишняя.

Решение задачи. Для примера на станции находятся порожние вагоны: полувагоны – 10, платформы – 10, крытые – 25. Все вагоны четырехосные грузоподъемностью 60 т. Заявка на погрузку: строительные грузы (щебень, гравий) – 800 т, глина – 400 т, лесные грузы – 300 т, торфяной брикет – 500 т.

Составляем начальный план: под погрузку 800 т строительных грузов назначаем 13 полувагонов, 400 т глины – 7 полувагонов, 300 т лесных – 8 платформ, 500 т торфяного брикета – 10 полувагонов. В табл. 4-8 приведены планы решений.

$$800/60=13,3 \text{ вагонов, проверка на перегруз } 800/13 - 60 = 1,5 \text{ т;}$$

$$400/60=6,6 \text{ вагонов, проверка на перегруз } 400/6 - 60 = 6,6 \text{ т;}$$

$300/39=7,7$  вагонов, проверка на перегруз  $300/7 - 39 = 3,8$  т;

$500/50 = 10$  вагонов.

Таблица 4

	Строительные 800	Глина 400	Лесные 300	Торфяной брикет 500	$\Sigma$	$\pm$
Пв 10	60 $\frac{13}{800}$	60 $\frac{7}{400}$	37	50 $\frac{10}{500}$	30	-20
Пл 10	45	35	39 $\frac{8}{300}$	0	8	+2
Кр 25	0	50	34	40	0	+25
$\lambda$	1,33	1,20	–	1,25		

В первой строке матрицы общая потребность (30 вагонов) превышает наличие (10 вагонов). Поэтому строка считается недостаточной с недостатком 20 вагонов. Вторая и третья строки избыточные с избытком соответственно 2 и 25 вагонов. Поскольку есть одна недостаточная строка, продолжим решение.

Подсчитаем разрешающие множители  $\lambda_j$  для каждого столбца с заполненными элементами в недостаточных строках делением максимальной технической нормы загрузки на максимальную норму в тех элементах этого же столбца, которые принадлежат избыточным строкам. В первом столбце матрицы максимальная техническая норма загрузки 60, в элементах избыточных строк этого столбца она равна 45. Поэтому разрешающий множитель для первого столбца  $\lambda_1 = \frac{60}{45} = 1,33$ , для второго  $\lambda_2 = \frac{60}{50} = 1,2$ . В третьем столбце элемент недостаточной строки не заполнен. Поэтому для него разрешающий множитель не вычисляется. Для четвертого столбца  $\lambda_4 = \frac{50}{40} = 1,25$ .



Преобразуем матрицу. Выбираем минимальное значение  $\lambda_j$  и на него либо умножаем в избыточных (если их меньше, чем недостаточных), либо делим в недостаточных строках (если их меньше, чем избыточных) все значения  $p_{ij}$ . В результате в преобразованной матрице изменены значения  $p_{ij}$  в некоторых строках. В дальнейших расчетах принимают участие именно эти значения технических норм загрузки вагонов. Однако при определении необходимого количества вагонов для погрузки того или иного груза во всех случаях используют лишь первоначальные значения  $p_{ij}$  (исходные данные).

В нашем случае минимальное значение у  $\lambda_2 = 1,2$ . Разделим на него все значения  $p_{ij}$  недостаточной строки (так как в матрице она одна). В преобразованной матрице значения  $p_{ij}$  в первой строке меньше исходных.

Таблица 5

	Строительные 800	Глина 400	Лесные 300	Торфяной брикет 500	$\Sigma$	$\pm$
Пв 10	50 $\frac{13}{800}$	50	31	42 $\frac{10}{500}$	23	-13
Пл 10	45	35	39 $\frac{8}{300}$	0	8	+2
Кр 25	0	50 $\frac{8}{400}$	34	40	8	+17
$\lambda$	1,11	–	–	1,05		

Выбираем столбец, в двух элементах которого одинаковые значения  $p_{ij}$ , принадлежащие избыточной и недостаточной строкам. Такому столбцу соответствует минимальное значение  $\lambda_j$ . Элемент избыточной строки называется альтернативным. Перенесем в этом столбце весь груз или часть его из элемента недостаточной строки в альтернативный, чтобы либо покрыть недостаток вагонов в данной недостаточной строке, либо использовать их

избыток в данной избыточной строке. рассчитаем количество используемых вагонов по первоначальному значению  $p_{ij}$ . Все остальные данные переносятся из предыдущей матрицы в преобразованную.

В новой матрице во втором столбце два элемента в избыточной (первой) и недостаточной третьей строке с одинаковой величиной  $p_{ij} = 50$ . Поскольку потребность в вагонах в этом столбце менее избытка и недостатка в соответствующих строках, перенесем весь груз из первой строки в третью. Однако теперь для погрузки уже потребуется не семь, а восемь вагонов, так как техническая норма загрузки снизилась с 60 до 50 т. Остальные элементы матрицы остаются без изменения. В результате недостаток вагонов в первой строке уменьшается, но все еще остается при наличии избыточных строк. Повторяем цикл расчетов, начиная с пункта 3. Подсчитываем разрешающие множители столбцов, у которых заполнены элементы в недостаточных строках. Минимальная величина разрешающего множителя  $\lambda_4 = 1,05$  находится в четвертом столбце. Преобразуем матрицу, поделив на эту величину значения  $p_{ij}$  в элементах первой недостаточной строки. В новой матрице заполняем элемент четвертого столбца последней избыточной строки, перенося туда весь груз из элемента первой недостаточной строки этого же столбца. Число назначаемых под погрузку вагонов увеличивается с 10 до 12 от снижения технической нормы загрузки. Поскольку в полученной матрице все еще имеются избытки и недостатки вагонов, продолжаем расчет.

Во вновь преобразованной матрице переносим в первом столбце из первой строки во вторую количество груза, достаточное для покрытия недостатка вагонов в первой строке:  $4 \times 45 = 180$  т. Для оставшегося в элементе первой строки в элементе первой строки груза  $800 - 180 = 620$  т необходимо  $620/60 = 10$  вагонов. Вторая строка становится недостаточной с недостатком 2 вагона. Первая строка также считается недостаточной.

Следующее преобразование матрицы. Умножаем величины  $p_{ij}$  в третьей избыточной строке на минимальное значение  $\lambda_3 = 1,15$ . Затем частично

переносим груз в третьем столбце из второй недостаточной строки в третью избыточную, чтобы покрыть недостаток. Полученный план – оптимальный. Три крытых вагона избыточны. Средняя статическая нагрузка составляет 47,6 т. Лучшего распределения вагонов при данных условиях добиться невозможно.

Таблица 6

	Строительные 800	Глина 400	Лесные 300	Торфяной брикет 500	$\Sigma$	$\pm$
Пв 10	48 $\frac{13}{800}$	48	30	40	13	-3
Пл 10	45	35	39 $\frac{8}{300}$	0	8	+2
Кр 25	0	50 $\frac{8}{400}$	34	40 $\frac{12}{500}$	20	+5
$\lambda$	1,07	–	–	–		

Таблица 7

	Строительные 800	Глина 400	Лесные 300	Торфяной брикет 500	$\Sigma$	$\pm$
Пв 10	45 $\frac{10}{620}$	45	28	37	10	-0
Пл 10	45 $\frac{4}{180}$	35	39 $\frac{8}{300}$	0	12	-2
Кр 25	0	50 $\frac{8}{400}$	34	40 $\frac{12}{500}$	20	+17
$\lambda$	$\infty$	–	1,15	–		

Таблица 8

	Строительные 800	Глина 400	Лесные 300	Торфяной брикет 500	$\Sigma$	$\pm$
Пв 10	45 $\frac{10}{620}$	45	28	37	10	0
Пл 10	45 $\frac{4}{180}$	35	39 $\frac{6}{232}$	0	10	0
Кр 25	0	58 $\frac{8}{400}$	39 $\frac{2}{68}$	46 $\frac{12}{500}$	22	+3

## **4 Разработка информационной системы**

Для разработки информационной системы рационального использования топливных ресурсов тепловоза реализуем математическую модель задачи при помощи технологии кроссплатформенного программирования, в частности языка Java. Необходимо разработать алгоритм математической модели и внешний интерфейс информационной системы.

### **4.1 Алгоритм**

Функционирования информационной системы осуществляется согласно структурному алгоритму, построенному на основе метода разрешающих множителей линейного программирования. Алгоритм изображен на рисунке 2.

В первую очередь заносится количество порожних вагонов каждого вида. Виды вагонов со своими характеристиками хранятся в отдельной базе данных. Предусматривается возможность редактировать характеристики вагонов.

Затем заносится количество и род груза. Род груза со своими характеристиками хранится в отдельной базе данных. Предусматривается возможность редактировать характеристики груза.

Далее идет реализация метода разрешающих множителей. Происходит поиск максимальной технической нормы погрузки. Затем вычисляется распределение груза по вагонам. При вычислении также происходит проверка на перегрузку. Затем рассчитывается итоговое суммирование вагонов.

Далее происходит проверка на оптимальность плана. Если план не считается оптимальным, то происходит вычисление разрешающего множителя. Затем в зависимости от недостатка или избытка вагонов в плане происходит коррекция технической нормы погрузки.

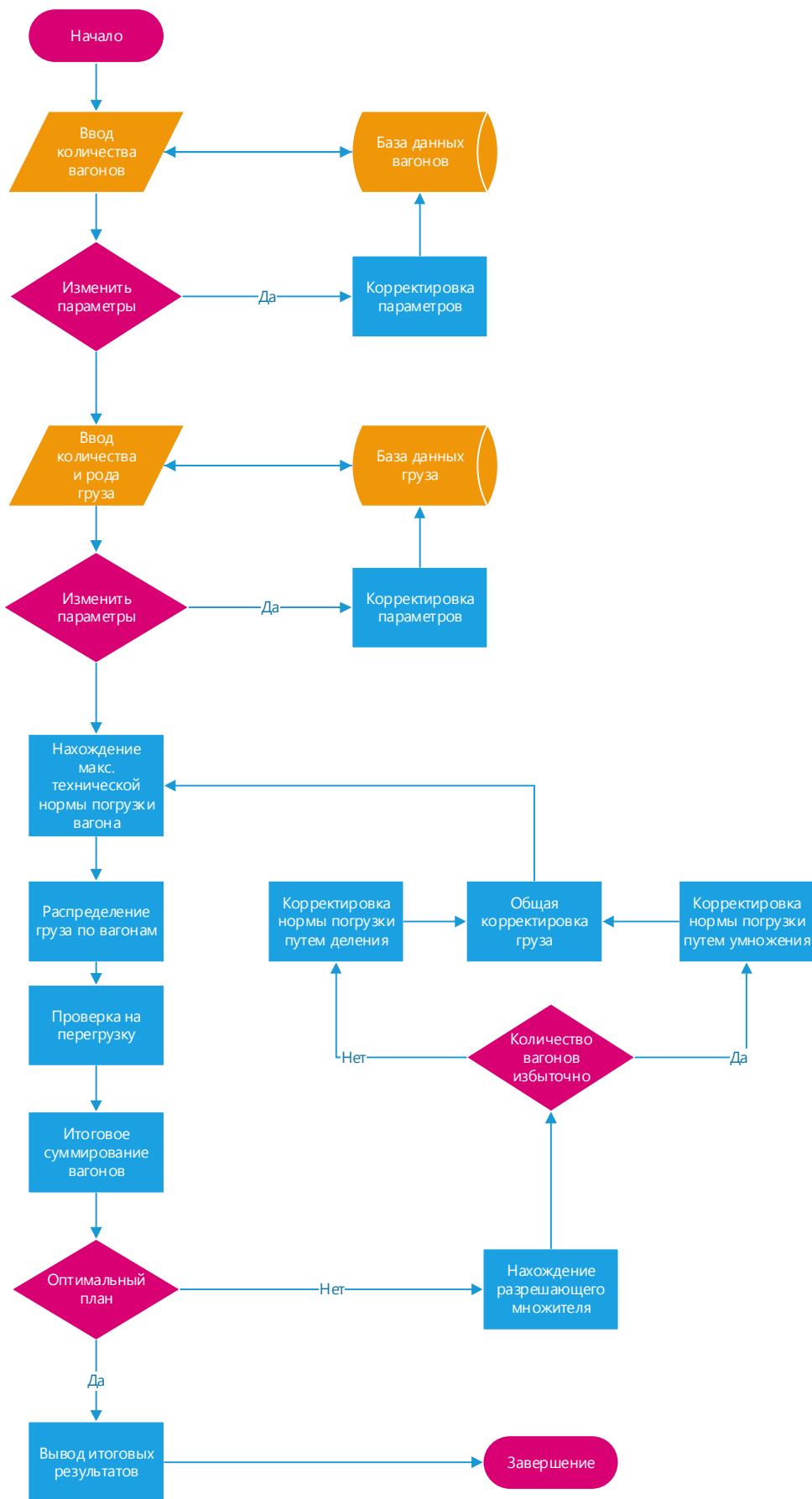


Рисунок 2 Алгоритм

Далее происходят общий пересчет распределения груза и возврат к нахождению максимальной технической погрузки.

Таким образом, итерация продолжается до тех пор, пока не будет получен оптимальный план. Далее результат распределения грузов по вагонам выводится на экран.

## 4.2 Выполнение программы

Информационная система оптимального использования топливных ресурсов представляет собой приложение с оконным интерфейсом, созданное с помощью языка Java и программной среды Eclipse.

Запускается приложение путем запуска файла «IS.exe». Экранные формы представлены на рисунках 3-7.

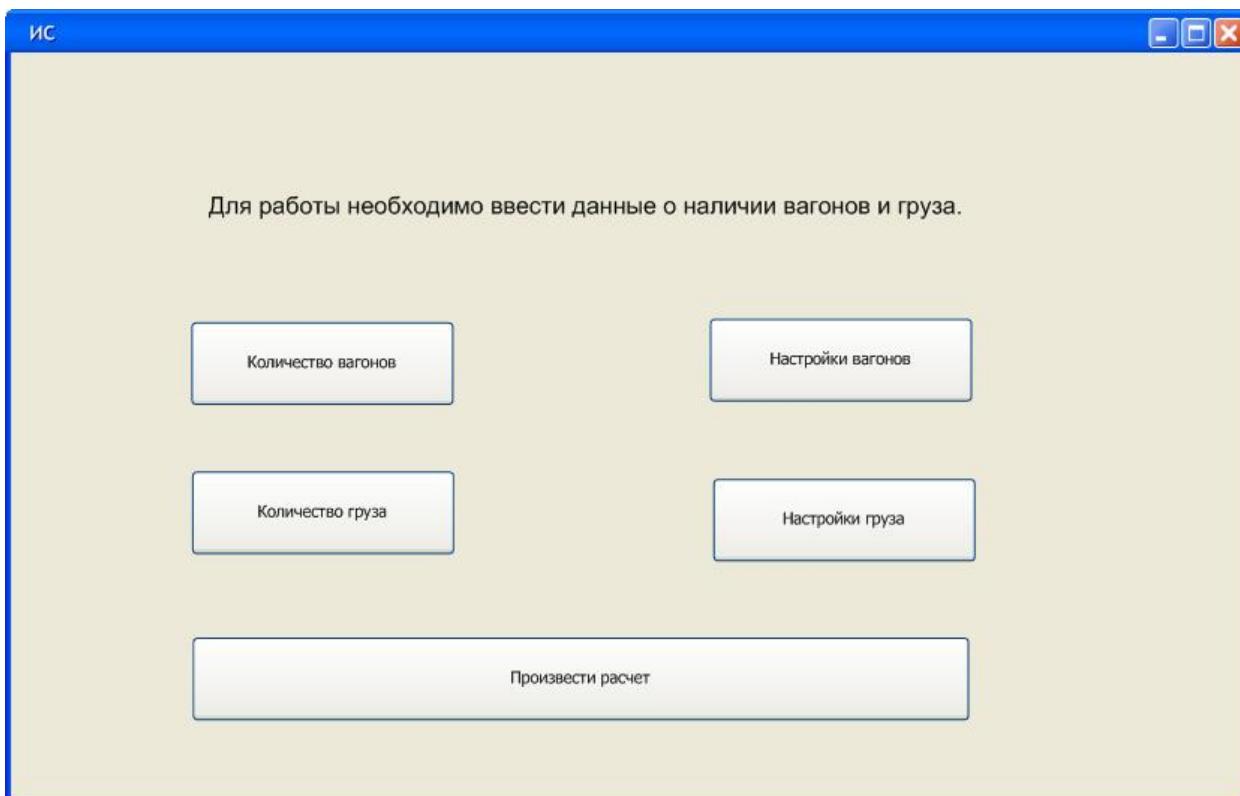


Рисунок 3 Главное окно программы

В главном окне программы пользователю предлагается ввести данные о вагонах и грузе, произвести настройки параметров вагонов и груза, а также произвести расчет оптимального распределения.

Окна ввода данных, а также настроек вагонов и груза открываются отдельными диалоговыми окнами.

Количество вагонов

Крытые	0	Думпкары	0
Платформы	0	Цистерны	0
Полувагоны	0	Изотермические	0
Хоппер	0	Рефрижератор	0

OK

Рисунок 4 Окно ввода количества вагонов

Настройка вагона

Тип	Грузоподъемность, т	Объем кузова, м <sup>3</sup>	Длина, м
Крытые	64	115	14
Платформы	66	37	14
Полувагоны	65	73	13
Хоппер	65	93	13
Думпкары	50	35	15
Цистерны	62	72	12
Изотермические	55	96	13
Рефрижератор	68	120	14

OK

Рисунок 5 Окно настройки характеристик вагонов



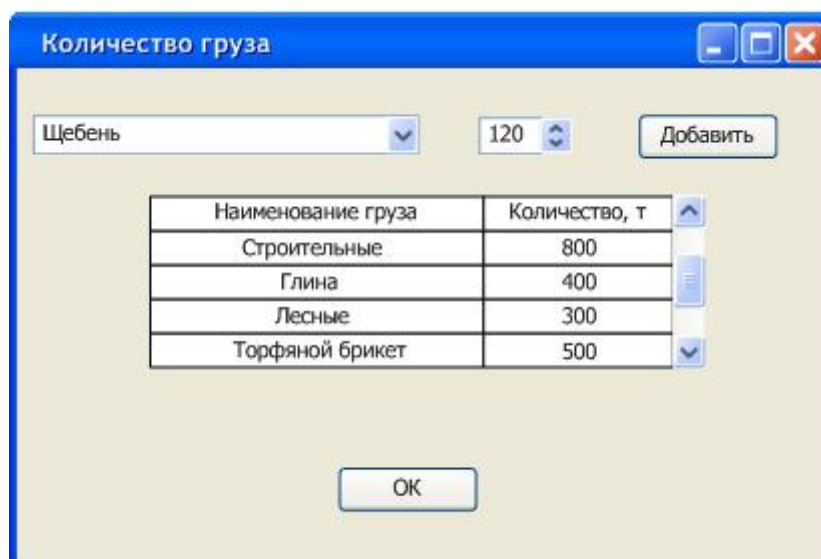


Рисунок 6 Окно ввода количества груза

По окончании ввода данных и нажатии кнопки «Произвести расчет» происходит расчет оптимального распределения грузов.



Рисунок 7 Окно итогового результата

По завершении расчета информация выводится на экран в виде таблицы. Информационная система позволяет распечатать результат, либо экспортировать данные в excel-файл.

При вводе данных в информационную систему, полученных из математической модели программа выдала результат, совпадающий с аналитическим решением.

На основе работы можно сделать вывод, что данная информационная система выполняет следующие задачи:

- позволяет вводить данные по количеству вагонов и груза;
- делает возможным для пользователя изменять характеристики вагонов и груза;
- производит вычисление оптимального количества вагонов;
- выводит результат вычисления в виде таблицы;
- позволяет пользователю распечатать результат или сохранить в виде файла;
- работает на различных операционных системах (в частности на Windows и Linux).

#### 4.3 Анализ информационной системы

Для того, чтобы продемонстрировать эффективность данной информационной системы представляется необходимым сделать сравнительный анализ. Анализ будет проводиться по следующим критериям:

- затраченное время на решение аналитическим путем, с помощью информационной системы и оператором;
- результат, полученный аналитическим путем, с помощью информационной системы и оператором.

Результаты анализа приведены в табл. 9

Таблица 9

Критерий	Оператор	Аналитическое решение	Информационная система
Время, мин.	11	27	4
Результат, вагонов	25	22	22

Оператор не владеет методами оптимального вычисления и решает задачу исходя из данных по грузам. Для решения аналитическим путем оператору необходимо владеть методами оптимального вычисления. В этом

случае достигается оптимальный результат, приводящий к экономии вагонов и топливных ресурсов тепловоза. Но в связи с этим требуется больше времени для произведения расчета и использование интеллектуальных способностей оператора.

Решение при помощи информационной системы позволяет избежать недостатков аналитического и простого решений. Для работы в информационной системе оператору не нужно владеть методами оптимального вычисления и использовать интеллектуальные способности. Также значительно уменьшается время на расчет результата. Что доказывает эффективность данной информационной системы.

## Заключение

В данной работе была разработана информационная система оптимального использования топливных ресурсов тепловоза при помощи технологии кроссплатформенного программирования. Были изучены методы математического программирования, произведено аналитическое решение задачи при помощи оптимального метода вычислений, изучен язык Java, разработан алгоритм аналитического решения задачи.

Данная информационная система выполняет следующие задачи:

- позволяет вводить данные по количеству вагонов и груза;
- делает возможным для пользователя изменять характеристики вагонов и груза;
- производит вычисление оптимального количества вагонов;
- выводит результат вычисления в виде таблицы;
- позволяет пользователю распечатать результат или сохранить в виде файла;
- работает на различных операционных системах (в частности на Windows и Linux).

При помощи сравнительного анализа было установлено, что данная информационная система позволяет:

- сократить время на проведение расчетов;
- уменьшить количество, используемых в работе, вагонов;
- автоматизировать вычислительные процессы.

На основе проведенного исследования можно сделать вывод, что при помощи информационной системы оптимального распределения грузов по вагонам можно добиться рационального использования топливных ресурсов тепловоза.

Данная информационная система может использоваться на предприятиях, осуществляющих перевозочную деятельность.

## Список литературы

1. Акулиничев В.М. Математические методы в эксплуатации железных дорог. М.: Транспорт, 1981.
2. Акулич И.Л. Математическое программирование в примерах и задачах. М.: Высш. шк., 1986.
3. Шилдт Г. Java 8. Полное руководство. М.: ООО «И.Д. Вильямс», 2015.
4. Ефименко Ю.И. Общий курс железных дорог. М.: Издательский центр «Академия», 2005.
5. Fain Y. Java Programming 24-Hour Trainer, Second Edition. Wiley Publishing, Inc., 2015.
6. Волкова В.Н., Горелова Г.В., Козлов В.Н., Фирсов А.Н. Моделирование систем и процессов. М.: Издательство Юрайт, 2014.
7. Абрамов Л. М., Капустин В.Ф. Математическое программирование. — Учебное пособие. — Л.: ЛГУ, 1981.