**ВВЕДЕНИЕ**

Развитые страны начали осваивать экологически чистую технологию безотходного производства. В России дело обстоит иначе. По сравнению с утилизацией на полигонах, безотходная промышленность обходится дороже. При этом в стране слабая законодательная база, а инфраструктура полностью отсутствует. Данное положение усугубляется тем, что во всех регионах страны процветают несанкционированные свалки.

Мусорная реформа

**1. ПРЕДМЕТНАЯ ОБЛАСТЬ**

* 1. **Понятие и сущность утилизации**

*Утилизация* – это уничтожение или переработка отходов для дальнейшего полезного использования в промышленности или быту. Непременным условием ее является безопасность. В результате получают энергию, топливо, сырье, материалы, изделия и т.п.

Утилизация играет важную роль в экономической цепочке производства. Она снижает расходы за счет повторного использования – рециклинга. Процент остатков, не пригодных для дальнейшего использования, таким образом сводится до минимума

Официальное определение термина дано в Федеральном законе от 24.06.1998 N 89-ФЗ .

Росприроднадзор считает целью утилизации не повторное использование, а уменьшение массовой доли мусора, снижение вредного воздействия на окружающую среду и здоровье человека.

*Все отходы сортируют на категории:*

* по классам опасности – с 1-го по 5-й;
* по агрегатному состоянию – жидкие, твердые, газообразные;
* по отраслевой принадлежности – бытовые, химические, строительные, технические, сельскохозяйственные, медицинские, военные и пр.;
* по материалу – металл, пластик, бумага, картон, стекло, ветошь, древесина, кирпич, пищевые отходы, нефтепродукты, резина, сточные воды и пр.;
* по происхождению – органические, неорганические.

*Виды переработки отходов*

* уничтожение;
* рециклинг;
* рекуперация;
* переработка.

Существующие опасные отходы подразделяются на группы. Для мусора каждого вида разработан специальный алгоритм, позволяющий минимизировать негативное воздействие на окружающую среду.

**Выделяется 4 блока происхождения ТБО:**

*- Химические*представляют собой определенное вещество или их совокупность в разном агрегатном состоянии. Могут быть инертными либо представлять угрозу человеческому здоровью и окружающей среде*;*

*- Минеральныеобразуются* в результате производственных и технологических процессов, при которых применяются минеральные вещества. Еще к данной категории относят изделия с минеральными составляющими, с истекшим сроком эксплуатации или поврежденные*;*

*- Органические*понимаются отходы растительного и животного происхождения, которые подвергаются разложению и гниению с образованием питательных веществ. Главными источниками являются пищевая, сельскохозяйственная сферы производства*;*

*- Коммунальные*представляют собой остатки потребления и производства, утратившие прежние свойства в процессе эксплуатации. Сбор, хранение и утилизация ТКО является законодательно регулируемым процессом*.*

Все ТКО делят на основе классификации классов опасности.

Класс опасности – важнейший критерий классификации мусора. ТБО делят на виды в зависимости от степени наносимого вреда окружающей среде и организму человека.

*Классы опасности:*

**1 класс – чрезвычайно опасные;**  
Это объясняется тем, что такие вещества не разлагаются, загрязняя ресурсы. К этой категории относят радиоактивные остатки, продукты, содержащие ртуть, синтетические масла.

**2 класс – высокой опасности;**  
Период нейтрализации опасных веществ длится не менее 30 лет.

**3 класс – средний уровень угрозы;**  
Нейтрализация негативного воздействия длится 10 лет. К данной категории относятся некоторые продукты сельского хозяйства, в том числе помет птиц, свиной навоз, дизельное топливо, очистной шлам.

**4 класс – малоопасные отходы с низкой степенью наносимого вреда;**  
Составляет, преимущественно, строительный и бытовой несортированный мусор, остатки производства продовольственных товаров.

**5 класс – практически не опасные.**  
Не оказывают пагубного воздействия на окружающую среду.

*Поэтому проблема утилизации отходов в России остается до сих пор на высоком уровне.*

*Процесс а организации бизнес процессы*

**2.Проектная часть**

**2.1Выбор средств и методов программирования**

Проектирование программного обеспечения – это процесс создания проекта программного обеспечения,

Выделяют следующие методы проектирования программных продуктов: структурный, объектно-ориентированный и процедурный.

Для разработки программного продукта будет применён объектно-ориентированный подход.

Объе́ктно-ориенти́рованное программи́рование — методология программирования, основанная на представлении программы в виде совокупности объектов, каждый из которых является экземпляром определённого класса, а классы образуют иерархию наследования.

При разработке программного обеспечения выделят следующие этапы:

* 1. Проектирование приложения
  2. Реализация программного кода приложения
  3. Тестирование приложения

На этапе проектирования системы планируется использовать унифицированный язык моделирования UML, использующийся для создания объектно-ориентированных моделей документации для программной системы. Для построения UML-диаграмм я остановил свой выбор на среде разработки StarUML. StarUML – инструмент моделирования программного обеспечения с открытым исходным кодом, обеспечивает одиннадцать типов диаграмм, StartUML 2 совместим с версиями UML 2.x.

Для реализации программного кода приложения планируется использовать высокоуровневый язык программирования Java, для создания графического интерфейса приложения. Средствами для реализации интерфейса служат библиотека JavaFX, среда разработки Intellij IDEA Community Edition и SceneBuilder.

Java – язык программирования общего назначения. Относится к объектно-ориентированным языкам программирования с сильной типизацией. Создатели реализовали принцип WORA: write once, run anywhere или «пиши один раз, запускай везде». Это значит, что написанное на Java приложение можно запустить на любой платформе, если на ней установлена среда исполнения Java (JRE, Java Runtime Environment).

Эта задача решается благодаря компиляции написанного на Java кода в байт-код. Этот формат исполняет JVM или виртуальная машина Java. JVM — часть среды исполнения Java (JRE). Виртуальная машина не зависит от платформы. Разработка ведёться через Java Community Process; язык и основные реализующие его технологии распространяются по лицензии GPL.

Intellij IDEA Community Edition - это интеллектуальная IDE, учитывающая контекст. Она предназначена для разработки разнообразных приложений на Java и других языках JVM, например Kotlin, Scala и Groovy.

Java FX –  это набор инструментов для создания кроссплатформенных приложений на основе языка программирования Java. Этот инструментарий был создан с целью упростить написание приложений. Другими словами, JavaFX — это декларативный язык сценариев со статической типизацией.

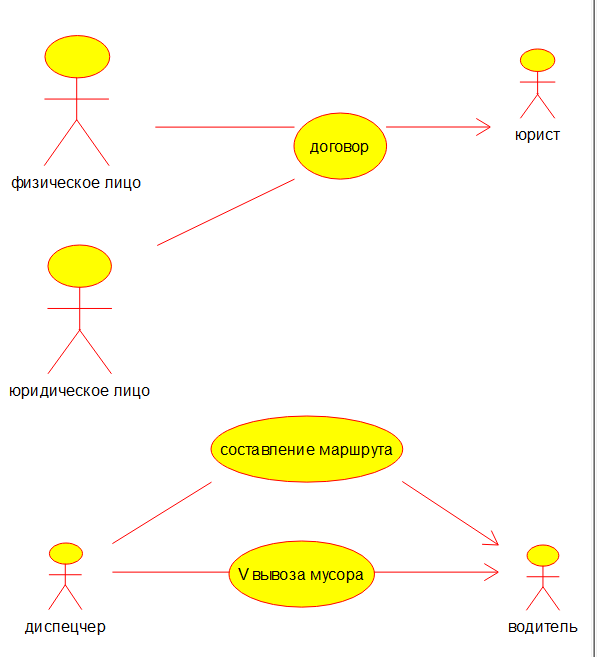
Scene Buider – это инструмент, с помощью которого можно конструировать окна в виде графического интерфейса и после их сохранять. Эта программа на основании результата будет конструировать xml файлы, которые мы будем редактировать в нашем приложении.

База данных, работающая с приложением будет проектироваться в MySQL Workbench. Это инструмент для визуального проектирования баз данных, интегрирующий проектирование, моделирование, создание и эксплуатацию базы данных в единое бесшовное окружение для данных MySQL.

Связь базы данных с интерфейсом обеспечивает драйвер JDBC и Connector/J. Это библиотеки для работы с базой данных, JDBC – общее название драйверов, которые используются в Java для работы с базой данных, а Connector/j – драйвер java для работы с MySQL.

**2.2 Абстрагирование и выделение классов**

**2.3Моделирование иерархий классов**

**2.4Построение диаграммы использования**

**2.5Построение диаграмм потоков данных и действий**

**3.1** **Разработка структур программного продукта**

Программное обеспечение для решения поставленных задач было реализовано в среде разработки IntelliJ IDEA на языке JavaScript . Данная среда разработки позволяет создать многофункциональное приложение .

Проект состоит из 2 основных классов таких как From, Login ….

Подробное описание классов в таблице 3.1

**3.2** **Проектирование классов сущностей**

**3.3** **Проектирование управляющих классов**

**3.4** **Проектирование интерфейсных классов**

**4 Экспериментальная часть**

