**КОМИТЕТ ПО ОБРАЗОВАНИЮ ПРАВИТЕЛЬСТВА САНКТ-ПЕТЕРБУРГА**

**САНКТ- ПЕТЕРБУРГСКОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ПРОФЕССИОНАЛЬНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ**

**“КОЛЛЕДЖ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ”**

Специальность 09.02.03 «Программирование в компьютерных системах»

**УТВЕРЖДАЮ**

**Зам. директора по УМР**

**\_\_\_\_\_\_\_\_\_\_\_\_ Венедиктова О.Н.**

**«\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2017 г.**

**ДИПЛОМНЫЙ ПРОЕКТ**

Пояснительная записка

**Тема: Симулятор распространения света**

Группа 443

Рецензент

Консультант по экономической части Т.В. Львова

Руководитель дипломного проекта А.В. Фомин

Выполнил Г.И. Зарипов

ОГЛАВЛЕНИЕ

[ОГЛАВЛЕНИЕ 2](#_Toc483072312)

[ВВЕДЕНИЕ 6](#_Toc483072313)

[1. АНАЛИТИЧЕСКАЯ ЧАСТЬ 8](#_Toc483072314)

[1.1. Исследование предметной области 8](#_Toc483072315)

[1.2. Обзор и анализ аналогичных решений 9](#_Toc483072316)

[1.2.1. Компьютерные версии 9](#_Toc483072317)

[1.2.2. Мобильные версии 10](#_Toc483072318)

[1.3. Анализ средств разработки 13](#_Toc483072319)

[1.3.1. Обзор методов решения 13](#_Toc483072320)

[1.3.1.1. Среда разработки 13](#_Toc483072321)

[1.3.1.2. Программы для разработки блок-схем и диаграмм 18](#_Toc483072322)

[1.3.1.3. Программа для разработки структуры базы данных 22](#_Toc483072323)

[1.3.1.4. Среда разработки аппаратной части 22](#_Toc483072324)

[1.3.1.5. Языки программирования 24](#_Toc483072325)

[1.3.2. Обоснование выбора методов 26](#_Toc483072326)

[1.3.2.1. Средства разработки 26](#_Toc483072327)

[1.3.2.2. Язык программирования 27](#_Toc483072328)

[1.3.2.3. Дополнительные средства 28](#_Toc483072329)

[1.3.2.4. Среда разработки аппаратной части 29](#_Toc483072330)

[2. ПРОЕКТНАЯ ЧАСТЬ 30](#_Toc483072331)

[2.1. Техническое задание 30](#_Toc483072332)

[2.1.1. Постановка задачи 30](#_Toc483072333)

[2.1.2. Назначение программного продукта 31](#_Toc483072334)

[2.1.3. Основание для разработки 31](#_Toc483072335)

[2.1.4. Требования к программе 31](#_Toc483072336)

[2.1.4.1. Требования к функциональным характеристикам 31](#_Toc483072337)

[2.1.4.2. Требования к аппаратным и программным средствам 31](#_Toc483072338)

[2.1.5. Требования к надежности 32](#_Toc483072339)

[2.1.5.1. Требование к надежной функциональности системы 33](#_Toc483072340)

[2.1.5.2. Отказ из-за некорректных действий пользователей системы 33](#_Toc483072341)

[2.1.6. Порядок контроля и приема 34](#_Toc483072342)

[2.1.7. Защита и сохранность данных 34](#_Toc483072343)

[2.2. Описание функционально-логической структуры программы 34](#_Toc483072344)

[2.2.1. Диаграмма потоков данных (DFD) 34](#_Toc483072345)

[2.2.2. Функциональная схема программы 35](#_Toc483072346)

[2.2.3. Функциональные диаграмма работы программы (IDEF0) 36](#_Toc483072347)

[2.2.4. Диаграмма вариантов использования (UML) 37](#_Toc483072348)

[2.3. Описание физико-математической модели 39](#_Toc483072349)

[2.3.1. Астрономические объекты 39](#_Toc483072350)

[2.3.1.1. Right ascension и Declination 39](#_Toc483072351)

[2.3.1.2. Спектральный класс 42](#_Toc483072352)

[2.3.1.3. Класс яркости 44](#_Toc483072353)

[2.3.2. Юлианский календарь 45](#_Toc483072354)

[2.3.3. Работа с векторами 50](#_Toc483072355)

[2.3.4. Системы координат 55](#_Toc483072356)

[2.4. Логическая модель данных 56](#_Toc483072357)

[2.4.1. Структуры 56](#_Toc483072358)

[2.4.2. Структура база данных 70](#_Toc483072359)

[2.5. Разработка аппаратной части 81](#_Toc483072360)

[2.5.1. Алгоритм работы устройства 85](#_Toc483072361)

[2.5.2. Сборка устройства 86](#_Toc483072362)

[2.5.3. Калибровка устройства 87](#_Toc483072363)

[2.6. Описание алгоритмов 89](#_Toc483072364)

[2.6.1. Общие алгоритм работы программы 89](#_Toc483072365)

[2.6.2. Алгоритмы отдельных модулей 101](#_Toc483072366)

[2.6.2.1. Взаимодействие форм программы 101](#_Toc483072367)

[2.6.2.2. Алгоритмы для работы с Юлианским календарём 101](#_Toc483072368)

[2.6.2.3. Алгоритмы для расчёта местоположения астрономических объектов 107](#_Toc483072369)

[2.6.2.4. Алгоритмы для построения изображения и вывода на экран 116](#_Toc483072370)

[2.6.2.5. Алгоритмы для отображения неба 120](#_Toc483072371)

[2.7. Программная реализация 126](#_Toc483072372)

[2.7.1. Пользовательский интерфейс 126](#_Toc483072373)

[2.7.1.1. Главное окно приложения 126](#_Toc483072374)

[2.7.1.2. Панель перемотки времени 127](#_Toc483072375)

[2.7.1.3. Строка меню 128](#_Toc483072376)

[2.7.1.4. Форма «Настройка местоположения» 130](#_Toc483072377)

[2.7.1.5. Форма «Таблица координат» 131](#_Toc483072378)

[2.7.1.6. Форма «Журнал наблюдений» 132](#_Toc483072379)

[2.7.2. Тестирование и отладка программы 133](#_Toc483072380)

[2.7.3. Локализация программного обеспечения 136](#_Toc483072381)

[2.8. Инструкция пользователя 139](#_Toc483072382)

[2.8.1. Настройка местоположения 140](#_Toc483072383)

[2.8.2. Основные сценарии программы 141](#_Toc483072384)

[2.8.2.1. Выбор темы приложения 141](#_Toc483072385)

[2.8.2.2. Настройка темы приложения 141](#_Toc483072386)

[2.8.2.3. Создание собственной темы приложения 142](#_Toc483072387)

[2.8.2.4. Добавление заметки в журнал наблюдения 143](#_Toc483072388)

[2.8.2.5. Изменение информации в заметки журнала наблюдения 144](#_Toc483072389)

[2.8.2.6. Удаление записи из журнала наблюдений 145](#_Toc483072390)

[2.8.2.7. Создание резервной копии базы данных 145](#_Toc483072391)

[2.8.2.8. Восстановление базы данных из резервной копии базы данных 146](#_Toc483072392)

[2.8.2.9. Подключение аппаратного устройства 147](#_Toc483072393)

[2.8.3. Подключение аппаратной части 147](#_Toc483072394)

[2.8.3.1. Подключение библиотек 147](#_Toc483072395)

[2.8.3.2. Калибровка магнитометра 149](#_Toc483072396)

[2.8.3.3. Работа с устройством 155](#_Toc483072397)

[2.9. Анализ результатов работы 155](#_Toc483072398)

[3. ЭКОНОМИЧЕСКАЯ ЧАСТЬ 157](#_Toc483072399)

[3.1. Описание продукта 157](#_Toc483072400)

[3.2. Описание и расчет затрат на выполнение проекта 157](#_Toc483072401)

[3.3. Расчет расходов на заработную плату 159](#_Toc483072402)

[3.4. Расчет затрат на материалы 159](#_Toc483072403)

[3.5. Расчет амортизации 159](#_Toc483072404)

[3.6. Расчет электроэнергии 160](#_Toc483072405)

[3.7. Расчет сметы затрат 160](#_Toc483072406)

[3.8. Расчет предполагаемой прибыли 160](#_Toc483072407)

[ЗАКЛЮЧЕНИЕ 163](#_Toc483072408)

[СПИСОК ИСПОЛЬЗУЕМЫХ ИСТОЧНИКОВ 164](#_Toc483072409)

[Приложение А. Исходный текст программы 165](#_Toc483072410)

[Приложение Б. Рисунки 166](#_Toc483072411)

ВВЕДЕНИЕ

Оптика — одна из древнейших наук, тесно связанная с потребностями практики на всех этапах своего развития. Прямолинейность распространения света была известна народам Месопотамии за 5 тыс. лет до н. э. и использовалась в Древнем Египте при строительных работах. На всем протяжении развития человечество развитие оптики означало прогресс в науке, также оптика необходима людям с плохим зрением, на момент 2015 года 50% старше 14 лет испытывали проблемы со зрением. Сложность оптических систем в телескопах растет с каждым годом, в телескопостроении используют адаптивную оптику, которая подстраивается под входящий поток света и выдает более четкую картинку.

Наибольшее значение оптики имеет для расчёта и конструирования оптических приборов - от очковых линз до сложных объективов и огромных астрономических инструментов.

Работа на персональных компьютерах позволяет на порядок упростить разработку, распространение и использование научных материалов, повышает качество образования, которое соответствует сегодняшним реалиям.

Так же правильное и комплексное использование компьютерных технологий предоставляет студентам и педагогам более эффективно распределять свое время, реализовывать творческие начала.

Это ведет к повышению спроса на специалистов: офтальмологов, инженеров-оптотехников, микробиологов, оптометристов, астрономов, а также продавцов очков.

В следствии этого было решено разработать симулятор оптической системы, который упростит понимание работы света и позволит моделировать простейшие оптические системы.

Данный диплом подтверждает компетенции профессиональных модулей:

1. ПМ 01. Разработка программных модулей программного обеспечения для компьютерных систем:

* ПК 1.1. Выполнять разработку спецификаций отдельных компонент.
* ПК 1.2. Осуществлять разработку кода программного продукта на основе готовых спецификаций на уровне модуля.
* ПК 1.3. Выполнять отладку программных модулей с использованием специализированных программных средств.
* ПК 1.4. Выполнять тестирование программных модулей.
* ПК 1.5. Осуществлять оптимизацию программного кода модуля.
* ПК 1.6. Разрабатывать компоненты проектной и технической документации с использованием графических языков спецификаций.

1. ПМ 02. Разработка и администрирование баз данных:

* ПК 2.1. Разрабатывать объекты базы данных.
* ПК 2.2. Реализовывать базу данных в конкретной системе управления базами данных.
* ПК 2.3. Решать вопросы администрирования базы данных.
* ПК 2.4. Реализовывать методы и технологии защиты информации в базах данных.

1. ПМ 03. Участие в интеграции программных модулей:

* ПК 3.1. Анализировать проектную и техническую документацию на уровне взаимодействия компонент программного обеспечения.
* ПК 3.2. Выполнить интеграцию модулей в программную среду.
* ПК 3.3. Выполнить отладку программного продукта с использованием специализированных программных средств.
* ПК 3.4. Осуществлять разработку тестовых наборов и тестовых сценариев.
* ПК 3.5. Производить инспектирование компонент программного продукта на предмет соответствия стандартам кодирования.
* ПК 3.6. Разрабатывать техническую документацию.

1. АНАЛИТИЧЕСКАЯ ЧАСТЬ

В этом разделе указана цель разработки данного проекта; проведен анализ существующих аналогов; описаны средства, использованные для разработки дипломного проекта.

* 1. Исследование предметной области

Оптика — это учение о физических явлениях, связанных с распространением коротких электромагнитных волн, длина которых составляет приблизительно 10-5 -10-7 м. Значение этой области спектра электромагнитных волн связано с тем, что внутри нее в узком интервале длин волн от 400-760 нм лежит участок видимого света, непосредственно воспринимаемого человеческим глазом.

Оптика разделяется на геометрическую, физическую и физиологическую. Геометрическая оптика ставит вопрос о природе света, исходит из эмпирических законов его распространения и использует представление о световых лучах, преломляющихся и отражающихся на границах сред с разными оптическими свойствами и прямолинейных в оптически однородной среде. Её задача - математически исследовать ход световых лучей в среде с известной зависимостью показателя преломления n от координат либо, напротив, найти оптические свойства и форму прозрачных и отражающих сред, при которых лучи происходят по заданному пути.

Физическая оптика рассматривает проблемы, связанные с природой света и световых явлений.

Одна из важнейших традиционных задач оптики - получение изображений, соответствующих оригиналам как по геометрической форме, так и по распределению яркости решается главным образом геометрической оптикой с привлечением физической оптики. Геометрическая оптика дает ответ на вопрос, как следует строить оптическую систему для того, чтобы каждая точка объекта изображалась бы также в виде точки при сохранении геометрического подобия изображения объекту. Она указывает на источники искажений изображения и их уровень в реальных оптических системах.

Целью разработки данного проекта является создание прикладного веб приложения, позволяющего построить и свою оптическую систему с заданными параметрами с возможностью экспорта и импорта сцен.

Были поставлены следующие задачи:

* выполнить обзор и анализ существующих аналогов;
* рассмотреть виды линз;
* изучить принцип преломления света;
* найти математическую модель транспортировки света;
* использовать полученные знания для рендеринга сцены.
  1. Обзор и анализ аналогичных решений

В ходе анализа рынка были рассмотрены программы, разработанные для компьютеров. Также рассмотрены существующие системы позиционирования телескопа и их реализация.

* + 1. Компьютерные версии

1. **OpticStudio**

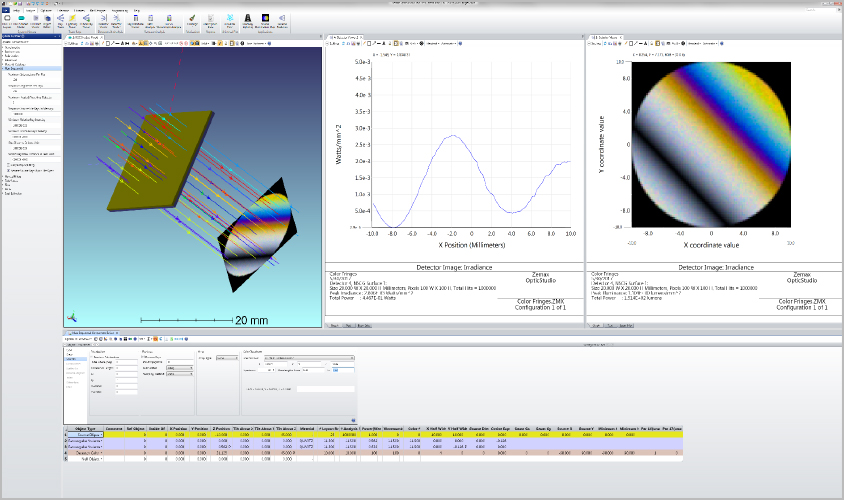


Рисунок 1 – Интерфейс OpticStudio

OpticStudio (рисунок 1) — это профессиональный инструмент для моделирования оптических приборов, обладающий широким функционалом.

Достоинства и недостатки приложения OpticStudio представлены в таблице 1.

Таблица 1 – Анализ приложения OpticStudio

|  |  |
| --- | --- |
| Достоинства | Недостатки |
| Огромное количество инструментов для анализа, моделирования и оптимизации оптических систем. | Высокая цена — от 5,5 тысяч долларов за персональную лицензию. |
| Разработчик постоянно улучшает программу и добавляет новый инструменты, которые идут в ногу с современными методами разработки оптических систем. |

Дополнительная информация о приложении представлена в таблице 2.

Таблица 2 – Информация о приложении OpticStudio

|  |  |
| --- | --- |
| Разработчик: | Zemax |
| Стоимость: | * $5,600 минимальная версия * $11,400 профессиональная версия * $17,100 премиальная версия * $8,940 годовая подписка на премиальную версию |
| Сайт разработчика: | https://www.zemax.com/ |
| Способы приобретения: | * официальный сайт; |

1. **LensMechanix**

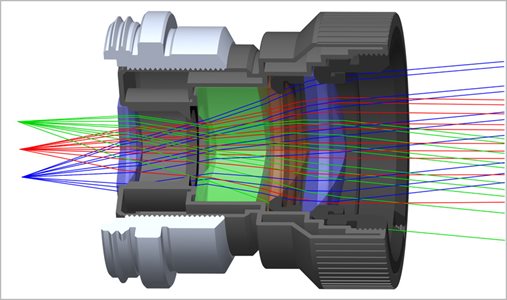


Рисунок 2 – Интерфейс LensMechanix

LensMechanix (рисунок 2) — это профессиональный инструмент для отладки оптических систем, работает в паре с OpticStudio.

Достоинства и недостатки ПО LensMechanix представлены в таблице 3.

Таблица 3 – Анализ приложения LensMechanix

|  |  |
| --- | --- |
| Достоинства | Недостатки |
| Наглядное представление прохождения лучей через линзу, а также есть возможность анимировать прохождение света через линзу. | Высокая цена — от 5,9 тысяч долларов. |
| Разработчик постоянно улучшает программу и добавляет новый инструменты, которые идут в ногу с современными методами разработки оптических систем. |

Дополнительная информация о приложении представлена в таблице 4.

Таблица 4 – Информация о приложении LensMechanix

|  |  |
| --- | --- |
| Разработчик: | Zemax |
| Стоимость: | * $5,900 стандартная версия |
| Сайт разработчика: | https://www.zemax.com/ |

1. **VirtualLab Fusion**

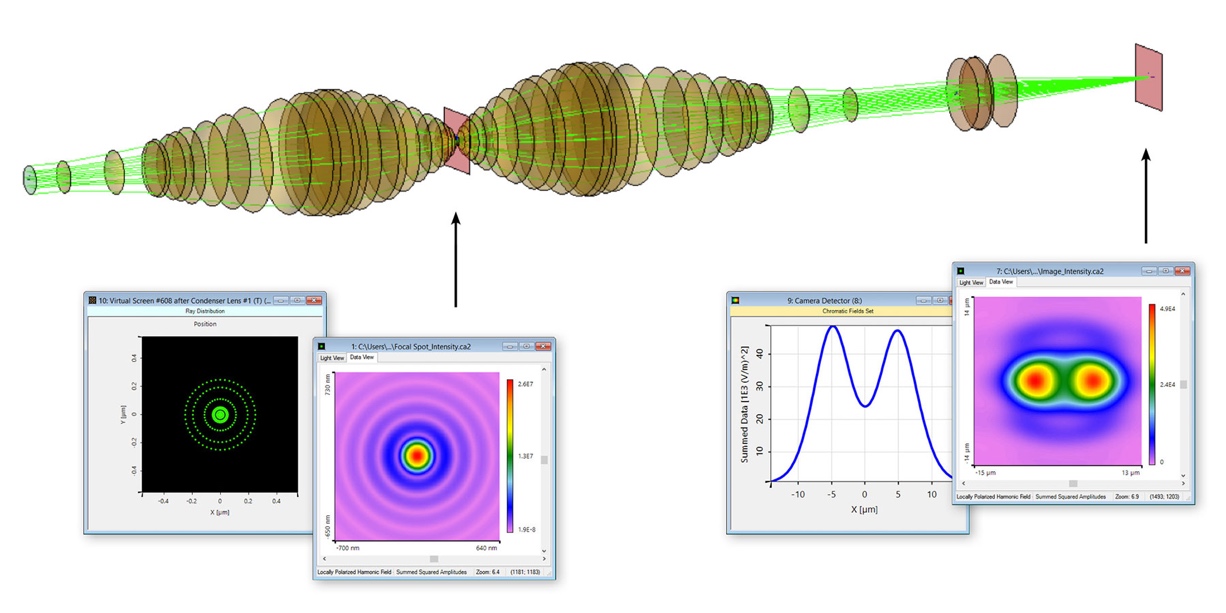


Рисунок 3 – Интерфейс VirtualLab Fusion

VirtualLab Fusion (рисунок 3) — это профессиональный инструмент для моделирования оптический систем.

Достоинства и недостатки ПО VirtualLab Fusion представлены в таблице 3.

Таблица 5 – Анализ приложения VirtualLab Fusion

|  |  |
| --- | --- |
| Достоинства | Недостатки |
| Есть документация к приложению, а также обучающие видеоролики. | Высокая сложность работы с ПО, а также договорная покупка продукта. |
| Разработчик постоянно улучшает программу и добавляет новый инструменты. |

Дополнительная информация о приложении представлена в таблице 6.

Таблица 6 – Информация о приложении VirtualLab Fusion

|  |  |
| --- | --- |
| Разработчик: | Wyrowski Photonics UG |
| Стоимость: | * Цены нет в публичном доступе |
| Сайт разработчика: | https://www.wyrowski-photonics.com |

* 1. Анализ средств разработки

Перед тем как приступать в разработке программы был произведен анализ средств разработки и языков программирования.

* + 1. Обзор методов решения

В данном разделе были рассмотрены средства разработки: среда разработки, программы для разработки блок-схем и диаграмм.

* + - 1. Среда разработки

Были проанализированы следующие среды разработки:

* Visual Studio Code
* Atom
* IntelliJ IDEA.

1. **Visual Studio Code**

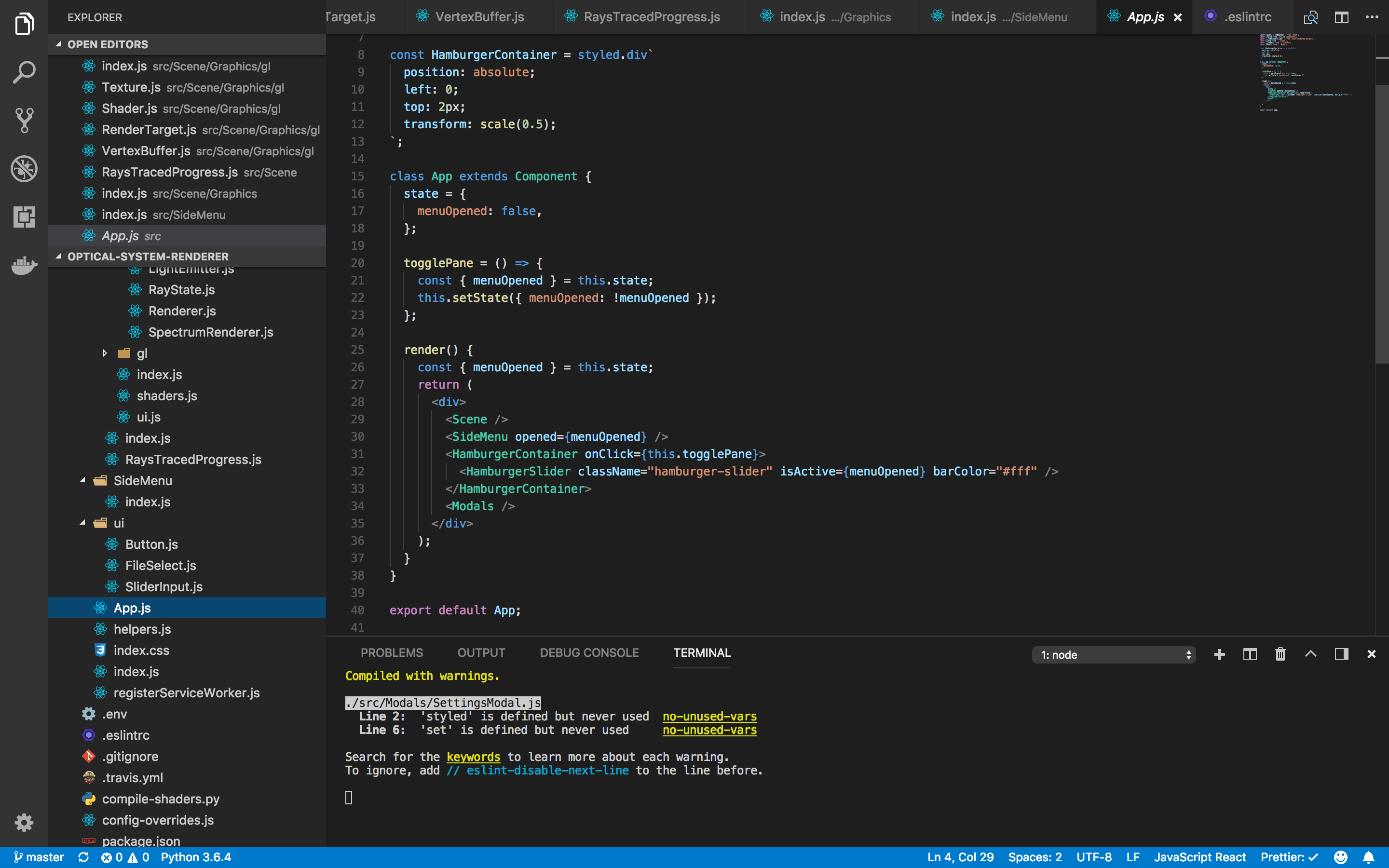


Рисунок 4 – Среда разработки Visual Studio 2015

Visual Studio Code (рисунок 4) – это бесплатная Integrated development environment (IDE[[1]](#footnote-1)), которая может быть использована индивидуальными разработчиками для создания собственных приложений, а также для проектов с открытым кодом, научных исследований и образования. Данная среда поддерживает большое количество языков: C++, C#, Visual Basic, F#, JavaScript, TypeScript, Python.

Данная среда имеет удобный и понятный интерфейс, что позволяет быстро приступить к разработке приложения.

Достоинства и недостатки приложения Visual Studio Code представлены в таблице 7.

Таблица 7 – Анализ приложения Visual Studio Code

|  |  |
| --- | --- |
| Достоинства | Недостатки |
| Является полностью бесплатной. | Не удобно сделан поиск по проекту. |
| Огромное количество плагинов, которые позволяют подстроить среду под ваши нужды |
| VS Code – это опенсорс IDE и имеет большое сообщество поклонников. |
| Быстрая скорость работы |

Дополнительная информация о приложении представлена в таблице 8.

Таблица 8 – Информация о приложении Visual Studio Code

|  |  |
| --- | --- |
| Разработчик: | Microsoft |
| Стоимость: | * Полность бесплатная |
| Сайт разработчика: | <https://www.visualstudio.com/ru/> |
| Способы приобретения: | * сайт разработчика. |

1. **Atom**

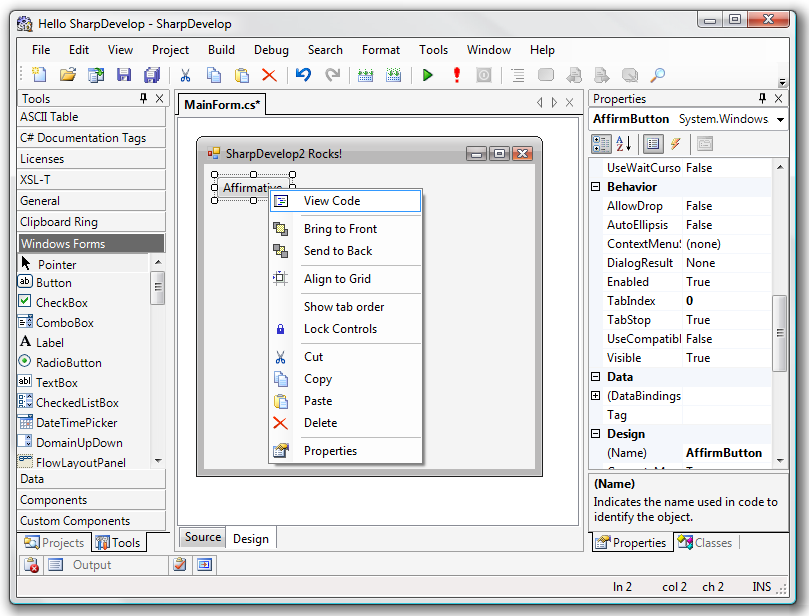


Рисунок 5 – Интерфейс Atom

Atom (рисунок 6) – кросс-платформенная среда разработки. Является полностью бесплатным. При первом же запуске очень сильно напоминает среду Visual Studio Code. Интерфейс довольно простой в освоении, что позволяет удобно разрабатывать приложения. Но стоит отметить то, что скорость работы значительно ниже, чем в Visual Studio Code.

Достоинства и недостатки приложения Atom представлены в таблице 9.

Таблица 9 – Анализ приложения Atom,

|  |  |
| --- | --- |
| Достоинства | Недостатки |
| Полностью бесплатная среда разработки. | Медленная скорость работы |
| Есть интеграция с github |
| Настроить можно любой элемент IDE. |

Дополнительная информация о приложении представлена в таблице 10.

Таблица 10 – Информация о приложении Atom

|  |  |
| --- | --- |
| Разработчик: | Atom |
| Стоимость: | * полностью бесплатная; |
| Сайт разработчика: | https://atom.io |
| Способы приобретения: | * сайт разработчика. |

1. **IntelliJ IDEA**

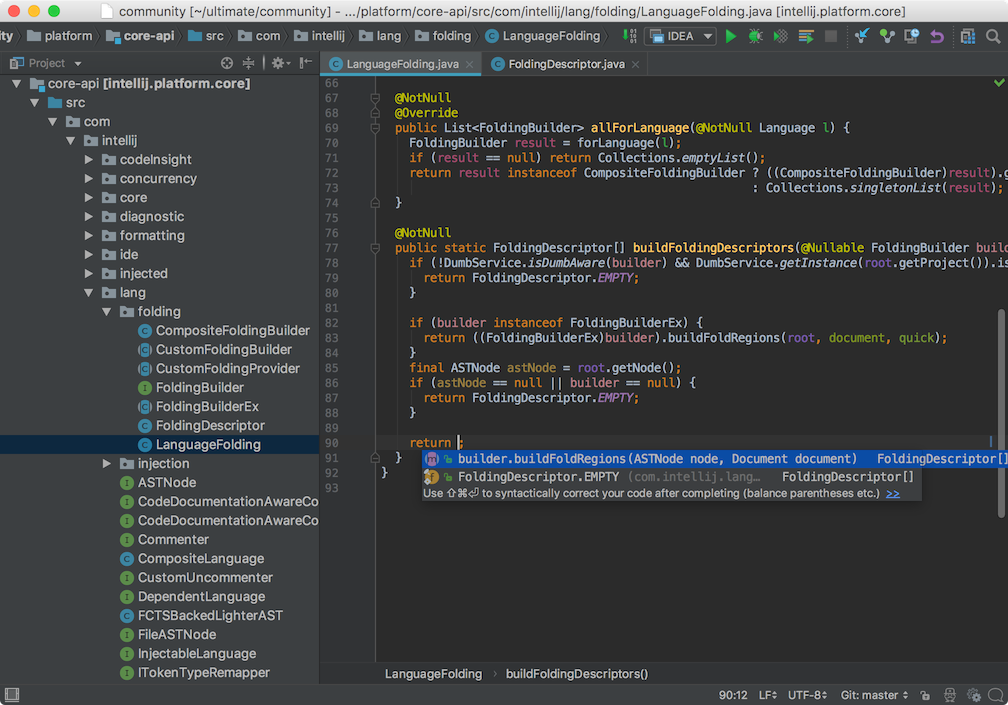


Рисунок 6 – Интерфейс IntelliJ IDEA

IntelliJ IDEA (рисунок 6) позволяет программировать на многих языках, в частности Java, Go, JavaScript, Python, Logtalk. Данная среда разработана компанией JetBrains. Появившись IntelliJ IDEA очень быстро стала завоёвывать свою часть рынка. Она предоставила пользователям удобный интерфейс, в котором нет лишних нагромождений. Среда постоянно помогает программисту напоминая об реализации обязательных методов или указывая ему на ошибки.

Достоинства и недостатки приложения IntelliJ IDEA представлены в таблице 10.

Таблица 11 – Анализ приложения IntelliJ IDEA

|  |  |
| --- | --- |
| Достоинства | Недостатки |
| Мощный редактор, помогающий на каждом шаге разработки приложений, начиная от подсветки и заканчивая напоминаниями об реализации нужных методов и рекомендациями в замене функций на более удобные выражения. | Среда несколько требовательна к ресурсам по сравнению с аналогами, но это окупается предоставляемым функционалом. |
| Возможность интеграции с Git и другими системами контроля версий. | Плохая поддержка JavaScript фреймворков. |
| Возможность построения диаграмм классов и переходы по иерархии (Ultimate Edition). | Долгая загрузка среды, для всех функция нужна платная версия. |

Дополнительная информация о приложении представлена в таблице 16.

Таблица 12 – Информация о приложении IntelliJ IDEA

|  |  |
| --- | --- |
| Разработчик: | JetBrains |
| Стоимость: | * бесплатная (Community); * платная (Ultimate). |
| Сайт разработчика: | <https://www.jetbrains.com/idea/> |
| Способы приобретения: | * сайт разработчика; * https://ru.uptodown.com. |

* + - 1. Программы для разработки блок-схем и диаграмм

Были проанализированы следующие программы для разработки блок-схем и диаграмм:

* Pencil Project;
* Gliffy;
* Diagram Designer;
* Draw.io;
* ConceptDraw.

1. **Gliffy**

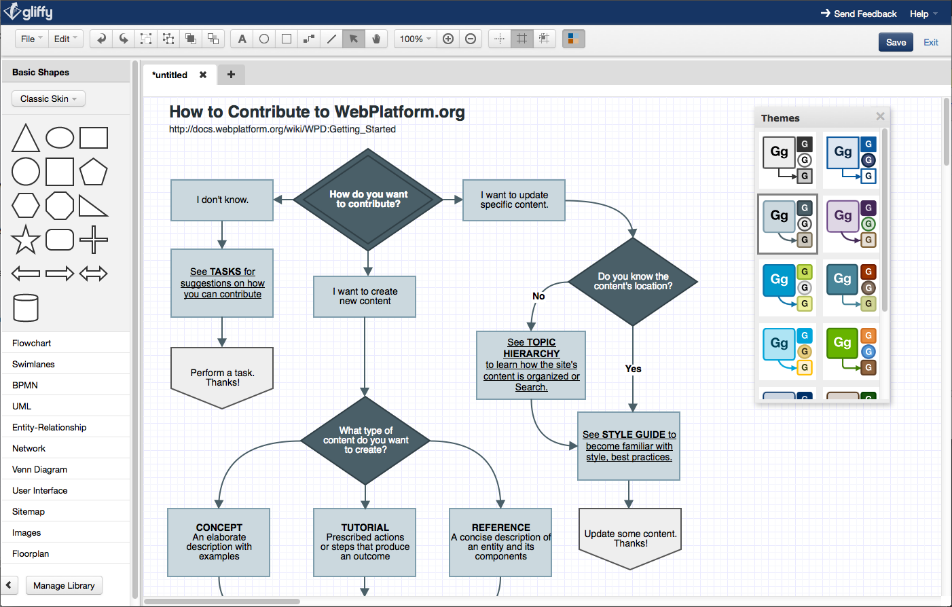


Рисунок 7 – Интерфейс Gliffy

Платное приложение, используемое для планирования и документирования, разработки программного обеспечения, бизнес-процессов и организационных диаграмм, с упором на создание функциональной деловой графики, например, диаграмм, блок-схем и каркасных представлений.

Данное приложение имеет легкий в использовании интерфейс, а также имеет большую базу стандартных форм, блоков и стрелок. Gliffy подходит для разработки: простых чертежей, блок-схем, UML диаграмм, ERD диаграмм и так далее.

Gliffy предоставляет пробную версию на 14 дней.

1. **Diagram Designer**

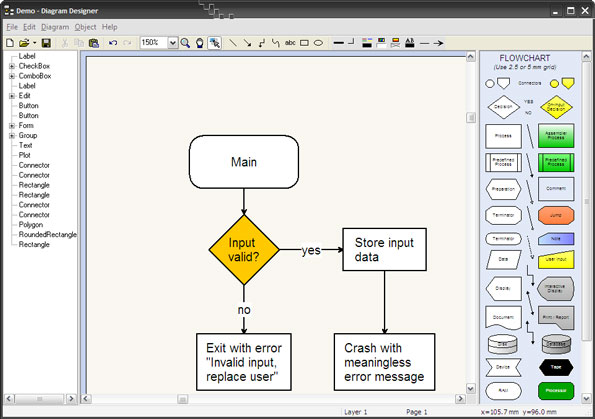


Рисунок 8 – Интерфейс Diagram Designer

Данная программа является полностью бесплатной и позволяет создавать диаграммы и презентации в векторе. Программа уже включает в себя базовые элементы и фигуры, которые могут быть использованы для построения диаграмм. Есть и газовые шаблоны, предназначенные для различных нужд. Но при этом стоит отметить, что в данной программу не включены все необходимые компоненты для самых известных блок-схем, принятых во многих документах.

1. [**Draw.io**](https://www.google.ru/url?sa=t&rct=j&q=&esrc=s&source=web&cd=1&ved=0ahUKEwjvjebnrojRAhUH3CwKHa5kASMQFggaMAA&url=https%3A%2F%2Fwww.draw.io%2F&usg=AFQjCNF2NBNfT4YkomteYAhnZgivowpxlg&sig2=lTXwpei4oPyO2l9fLfZEeQ&bvm=bv.142059868,d.bGg)

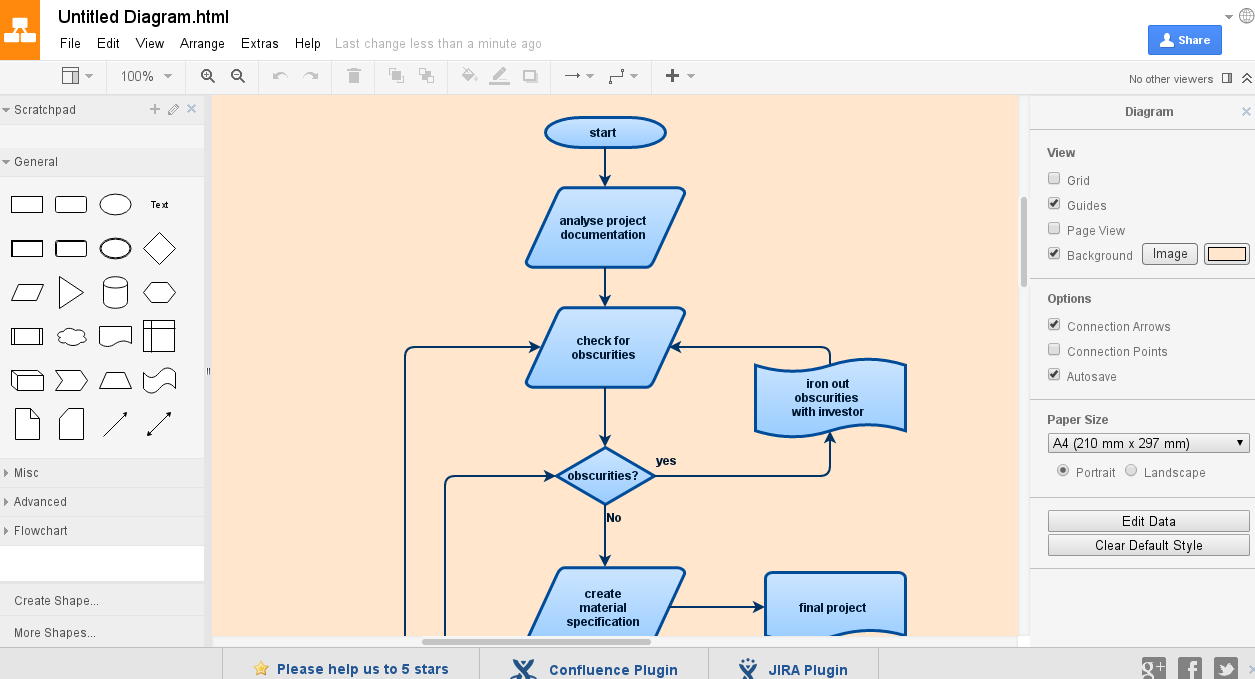


Рисунок 9 – Интерфейс Draw.io

Данное приложение имеет большой набор фигур предназначенных для рисования блок-схем, диаграмм и есть даже заготовки для рисования схем компьютерных сетей, баз данных и диаграмм UML; в настройках ты можешь включить и другие наборы — для электронных схем, интерфейсов iOS и Android, макетов сайтов.

Можно произвести настройку шрифта, обводки и заливки фигур. А также есть слои для распределения элементов диаграммы.

Draw.io поддерживает импорт файлов Visio, а сохранять умеет как в растровые форматы, так и в SVG.

1. **ConceptDraw**

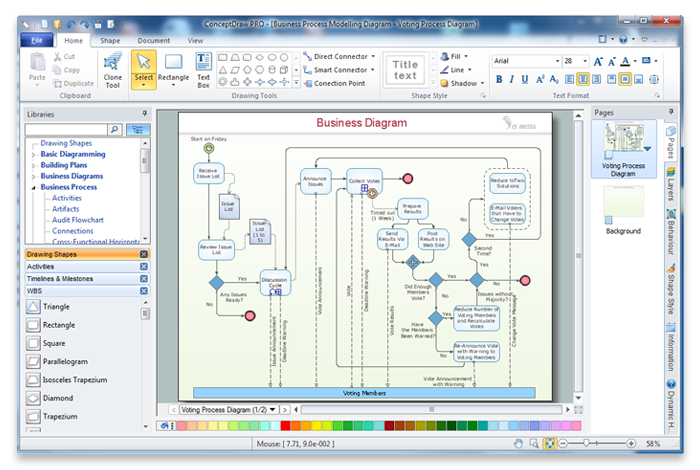


Рисунок 10 – Интерфейс ConceptDraw

Является полноценным и профессиональным приложением для разработки схем, чертежей и ментальных карт, но большая часть контента является платной, предоставляя пользователю узкий функционал. При этом стоит отметить, что внутри данной программы предоставлены различные шаблоны для большого количества областей: бизнес, маркетинг, разработка программных продуктов, графики, карты, качество, компьютеры и сети и многое другое. В каждой из областей заложены основные типы схем, которые чаще всего используются в жизни для оформления документов. Например, для разработки программных продуктов предложены шаблоны ERD, UML, IDF0, DFD диаграмм, а также есть возможность разработки эскиза приложения.

* + - 1. Языки программирования

1. **JavaScript**



Рисунок 11 – Пример кода на C#

Многопарадигменный, высокоуровневый, интепретирумый и динамический язык программирования, относящийся к семейству С-подобных языков.

JavaScript был создан для использования в браузере Netscape в 1995 году. С помощью него можно создавать веб-страницы с динамическим контентом, кроссплатформенные нативные приложения с помощью фреймворка Electron, серверные приложения с помощью платформы Node.js.

1. **C++**

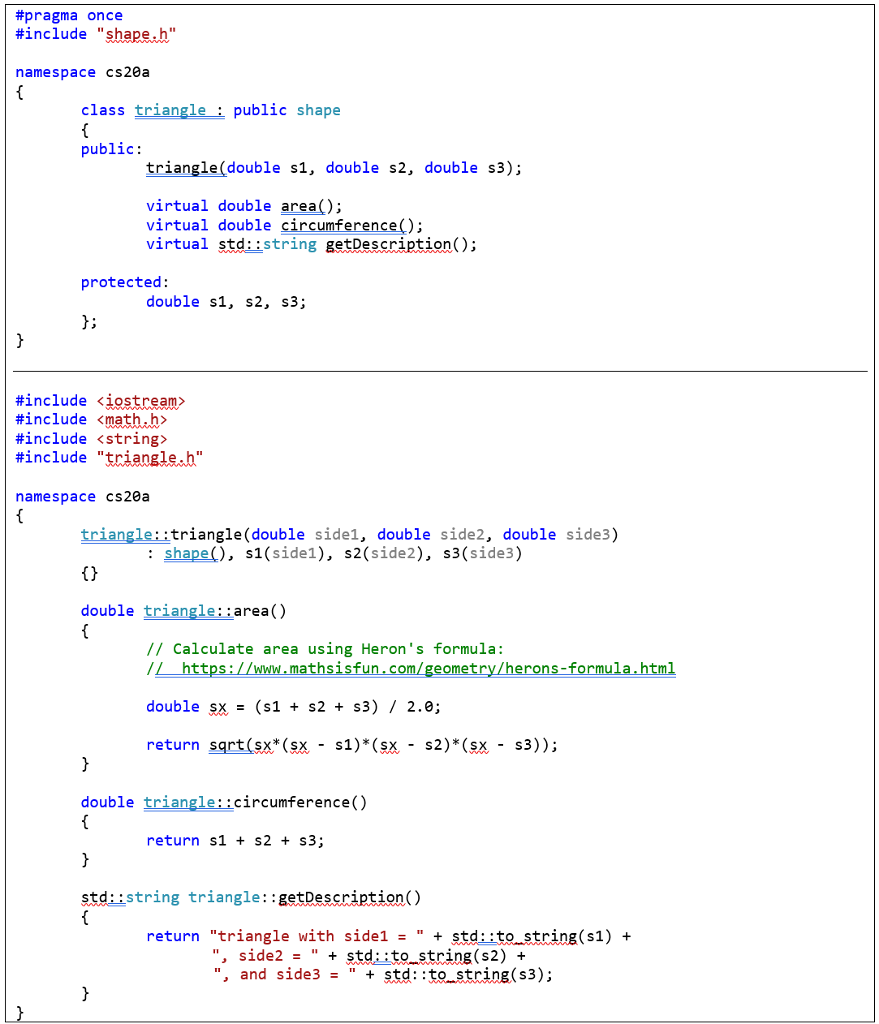


Рисунок 18 – Пример кода на C++

Компилируемый, статически типизированный язык программирования общего назначения. Является продолжением языка C, сделав акцент на объектно-ориентированное и обобщенное программирование. Сфера применения данного языка очень обширна: создание драйверов, операционных систем, серверов и т.д.

Хотя первоначально С++ был нацелен на работу с очень большими программами, это не ограничивает его применение. С++ используют тогда, когда готовы пожертвовать временем ради скорости работы программы. Этот язык часто используется для таких проектов, как создание редакторов, баз данных, персональных систем работы с файлами и коммуникационных программ. Благодаря тому, что С++ унаследовал эффективность языка С, с его помощью разрабатывается высокопроизводительное программное обеспечение.

1. **Java**

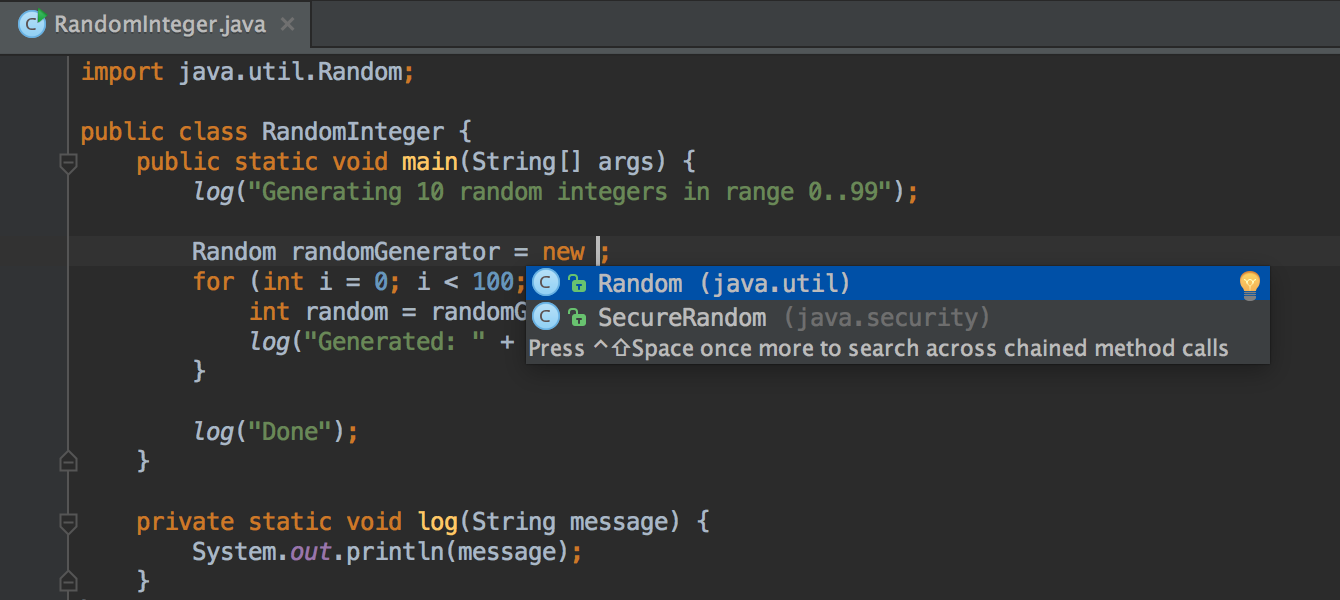


Рисунок 19 – Пример кода на Java

Структурированный, объектно-ориентированный язык с синтаксисом и конструктивными особенностями, унаследованными от С++ и Smalltalk. Java стал рассматриваться вопрос переносимости программ и путём получения промежуточного байт-кода из исходного кода это стало возможным. Это позволило решить вопрос перекомпиляции программы для каждой отдельной системы. Также переработали ООП, добавили сборщик мусора и синтаксического сахара. Java в основном используется для разработки больших Enterprise-приложений.

* + 1. Обоснование выбора методов

Проанализировав методы разработки были выбраны наиболее подходящие из них для реализации данного проекта.

* + - 1. Средства разработки

В ходе анализа в качестве среды разработки была выбрана Visual Studio Code.

Полнофункциональная интегрированная среда разработки для написания приложений для Windows, Android, MacOS и iOS, а также веб-приложений и облачных служб. Данная среда очень просто в использовании, что позволяет быстро и без особых проблем приступить к началу разработки проекта. Также Visual Studio Code имеет встроенный отладчик, который позволяет проанализировать производительность приложения; выявить ошибки; гибкая проверка состояний, которая позволяет просмотреть значение переменных во время выполнения программы.

Данная среда разработка не только позволяет что-то разрабатывать самому, но и пытается помочь программисту в ходе разработки, предоставляя автоформатирование и проверку стиля кода, а также при нахождении ошибок есть возможность выбрать один из предлагаемых вариантов от Visual Studio Code для её устранения.

Во время разработки приложения есть возможность проведения тестирования. Для этого можно создать перечень тестовых наборов, чтобы не писать их каждый раз.

Visual Studio Code является отличным решением для веб-разработки, так очень проста в использовании и при этом позволяет конфигурировать все что нужно, есть возможность расширения функционала с помощью установки плагинов, нативная поддержка Mac OS. В Visual Studio Code имеется отличная поддержка современных JavaScript фреймворков и библиотек. Удобная работа с системами контроля версий и гитом, в частности. Visual Studio Code – стандарт при написании веб-приложений и SPA (Single Page Application).

* + - 1. Язык программирования

В ходе анализа трёх языков C++, JavaScript и Java был выбран JavaScript.

Javascript был создан в 1995 году и стал единственным языком который поддерживают все браузеры. Данный язык вдохновлен языком Java, но структурно почти ничего не унаследовал.

JavaScript изначально создавался для того, чтобы сделать web-странички «живыми». Программы на этом языке называются скриптами. В браузере они подключаются напрямую к HTML и, как только загружается страничка, выполняются. Может выполняться не только в браузере, а где угодно, нужна лишь специальная программа – интерпретатор. Процесс выполнения скрипта называют «интерпретацией».

JavaScript – быстрый и мощный язык, но браузер накладывает на его исполнение некоторые ограничения. Это сделано для безопасности пользователей, чтобы злоумышленник не мог с помощью JavaScript получить личные данные или как-то навредить компьютеру пользователя. Этих ограничений нет там, где JavaScript используется вне браузера, например на сервере. Кроме того, современные браузеры предоставляют свои механизмы по установке плагинов и расширений, которые обладают расширенными возможностями, но требуют специальных действий по установке от пользователя.

Сам язык JavaScript улучшается. Современный стандарт ECMAScript 5 включает в себя новые возможности для разработки, ECMAScript 6 будет шагом вперёд в улучшении синтаксиса языка.

Современные браузеры улучшают свои движки, чтобы увеличить скорость исполнения JavaScript, исправляют баги и стараются следовать стандартам.

В итоге, можно сказать, что язык JavaScript уникален благодаря своей полной интеграции с HTML/CSS. Он работает почти у всех посетителей.

* + - 1. Дополнительные средства

В данном разделе пояснительной записки к дипломному проекту описываются дополнительные средства, используемые при разработке программного продукта.

1. **Gliffy**

Данное приложение показалось наиболее удобным в использовании, так как имело расширенный набор блоков и элементов для наиболее распространённых стандартов. Система очень легко передвигает блоки и соединяет их между собой, что позволяет быстро создать необходимую схему.

С помощью данного приложения были разработаны блок-схемы алгоритмов, структурная и функциональные схемы программы, а также диаграмма вариантов использования.

По сравнению с другими локальными программами, данное интернет приложение позволяет всё хранить и редактировать в сети, что позволяет работать над проектом не зависимо от местонахождения.

1. **ConceptDraw**

Данная программа позволило реализовать функциональную диаграмму работы программы. В приложении предусмотрены все типы стрелок: входные и выходные данные, управление и механизм. При этом можно удобно детализировать перенося все стрелки из контекстной диаграммы в детализированную.

1. ПРОЕКТНАЯ ЧАСТЬ

Проектная часть дипломного проекта содержит подробную информацию о разработанном программном продукте: назначение программы, требования, предъявляемые к самому приложению и техническим средствам, структурное построение программного продукта. Также в этом разделе описывается инструкция пользователя, назначением которой является обеспечение первоначального знакомства с программным продуктом.

* 1. Техническое задание

Перед тем как приступить к разработке программного продукта составим техническое задание.

Техническое задание (ТЗ) – исходный документ на проектирование технического объекта (изделия). ТЗ устанавливает основное назначение разрабатываемого объекта, его технические характеристики, показатели качества и технико-экономические требования, предписание по выполнению необходимых стадий создания документации и её состав, а также специальные требования.

* + 1. Постановка задачи

Задачей дипломного проекта является создание прикладного приложения, которое должно иметь следующие функции:

* отображение интерактивной оптической системы;
* добавление и удаление линз и источников света;
* сохранение и удаление сцены;
* настройка качества изображения;
  + 1. Назначение программного продукта

Созданная программа должна быть использовано как справочное пособие, которое позволит рассмотреть звезды и созвездия, прочитать про них информацию. Также предназначена для расчётов местоположения звездных объектов в зависимости от времени и места наблюдения.

* + 1. Основание для разработки

На рынке нет бесплатного аналога, который может красиво продемонстрировать преломление света в оптической системе.

* + 1. Требования к программе

В данной части описываются требования к программному обеспечению и аппаратных характеристик для правильного функционирования приложения.

* + - 1. Требования к функциональным характеристикам

Программа должна выполнять следующие функции:

* Отображать интерактивную оптическую систему;
* Рассчитывать преломления лучей;
* Показывать информацию о линзе;
  + - 1. Требования к аппаратным и программным средствам

Для работы данного приложения на компьютере необходимо установить один из следующих браузеров:

* Mozilla Firefox 4.0 и выше.
* Google Chrome 9 и выше.
* Opera 12.0 и выше.
* Apple Safari.

Рекомендуемые системные требования представлены в таблице 17:

Таблица 17 – Системные требования для установки программы

|  |  |
| --- | --- |
| Операционная система | * Windows, MacOS, Linux; |
| Процессор | Intel Core i3 или новее |
| Свободное место на диске | 30 МБ |
| Оперативная память | 256 МБ |

* + 1. Требования к надежности

Данное приложение не использует персональные данные пользователей или другие дополнительные конфиденциальные данные, в следствие этого отсутствует причина использования дополнительных методов защиты, таких как шифрование.

В этом проекте используются следующие средства:

* проверка файла при загрузке сцены;
  + - 1. Требование к надежной функциональности системы

Для надежного функционирования системы необходимо обеспечить проверку подключения файлов при запуске системы. При их отсутствии оповестить пользователя и предложить восстановить базу данных из резервной копии. В случаи критических ошибок функционирование системы предложить пользователю переустановить приложение.

Если ошибки не являются критичными для приложения, и оно может дальше функционировать, то оповестить о них пользователя и предложить дальнейшее выполнение действий, не обращая на них внимания.

* + - 1. Отказ из-за некорректных действий пользователей системы

Чаще всего ошибки возникают в ходе эксплуатации программного продукта из-за того, что пользователь ввёл неправильные данные или совершил непредсказуемые действия. Программа должна быть готова к таким действия и правильно работать независимо от ошибок.

В первую очередь необходимо обеспечить проверку данных, вводимых в поля программы. При неправильных входных данных, необходимо выдать оповещение о некорректности указанных данных и попросить пользователя ввести новые.

Также необходимо контролировать действия пользователя и сопровождать его на протяжении всего времени работы с приложением. Необходимо указывать на неполное заполнение полей, неправильный формат данных, невозможность выполнения действия с указанием причины. Так пользователь поймёт, что просит от него программа и сможет получить необходимые ему данные.

1. *IDE (Integrated development environment) –* комплекс программных средств, используемый программистами для разработки программного обеспечения. [↑](#footnote-ref-1)