**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«РОССИЙСКАЯ АКАДЕМИЯ НАРОДНОГО ХОЗЯЙСТВА   
И ГОСУДАРСТВЕННОЙ СЛУЖБЫ**

**ПРИ ПРЕЗИДЕНТЕ РОССИЙСКОЙ ФЕДЕРАЦИИ»**

**НИЖЕГОРОДСКИЙ ИНСТИТУТ УПРАВЛЕНИЯ – филиал РАНХиГС**

Факультет управления

Кафедра Информатики и информационных технологий

Направление подготовки / специальность: 09.04.03 Прикладная информатика

Направленность (профиль) / специализация: Корпоративные информационные системы управления

**Отчет**

|  |  |
| --- | --- |
| по дисциплине: | Программная инженерия |
|  |  |
|  |  |
|  |  |

|  |  |  |
| --- | --- | --- |
|  |  | **АВТОР** |
|  |  | Обучающийся(иеся) 2 курса группы Ик-721 |
|  |  | заочной формы обучения |
|  |  |  |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ Гулянцев А.Н. |
|  |  | *(подпись) (фамилия, инициалы)* |
|  |  |  |
|  |  | **ДОЛЖНОСТЬ ПРЕПОДАВАТЕЛЯ** |
|  |  | Старший преподаватель кафедры |
|  |  | *(ученая степень, ученое звание)* |
|  |  | оценка \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |
|  |  | «\_\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г. |
|  |  | *(дата защиты)* |
|  |  | \_\_\_\_\_\_\_\_\_\_\_\_\_ Окулич В.И. |
|  |  | *(подпись) (фамилия, инициалы)* |

Нижний Новгород, 2022г.

# Освоение использования системы GitHub. История, описание, скриншоты работы.

**История.**

GitHub — крупнейший[7] веб-сервис для хостинга IT-проектов и их совместной разработки.

Веб-сервис основан на системе контроля версий Git и разработан на Ruby on Rails и Erlang компанией GitHub, Inc (ранее Logical Awesome). Сервис бесплатен для проектов с открытым исходным кодом и (с 2019 года) небольших частных проектов, предоставляя им все возможности (включая SSL), а для крупных корпоративных проектов предлагаются различные платные тарифные планы.

Сервис для совместной разработки и хостинга ИТ-проектов Github был основан в апреле 2008 года программистами Крисом Ванстрасом (Chris Wanstrath), Томом Престон-Вернером (Thomas Preston-Werner) и Пи Джей Хайеттом (PJ Hyett). Первоначально проект предполагался как сообщество по разработке с помощью открытого исходного кода, использующего систему управления версиями Git.

В 2005 году программист Линус Торвальдс создал Git в ходе работы над ядром Linux. Разработчики начали применять эту систему в своей работе. Создание Git считалось выходом «на следующий уровень» по сравнению с другим программным обеспечением для управления версиями.

При этом, по мнению Престон-Вернера, проблема совместной работы не исчезла — Git делала её возможной, но сложной. GitHub был призван стать решением проблемы. В дальнейшем первым слоганом платформы станет «Гит-хостинг: больше не боль в заднице» (Git hosting: No longer a pain in the ass).

Идея пришла Ванстрасу и Престону-Вернеру во время митапа Ruby-разработчиков в октябре 2007 года в Сан-Франциско. Они планировали, что будущий проект должен стать местом, где разработчики могли бы легко делиться кодом и изучать программы друг друга.

В то время Ванстрас зарабатывал консалтингом, а до этого занимался техническим развитием сайта о технологиях Cnet. В свою очередь Престон-Вернер был разработчиком в компании Powerset.

В конце 2007 года они приступили к созданию собственного проекта — по субботам разработчики устраивали мозговой штурм, а в свободное время писали код будущей платформы. Престон-Вернер взял на себя разработку пользовательского интерфейса GitHub, а Ванстрас занялся разработкой самого приложения на Ruby on Rails.

«GitHub не должен был стать стартапом или компанией. GitHub был всего лишь инструментом, который был нужен нам самим», — рассказывал в интервью Wired Ванстрас.

Престон-Вернеру в работе над GitHub также пригодился опыт создания сервиса Gravatar в 2007 году, который позволял сохранять изображения для профилей в соцсетях на отдельном сервере. В том же году разработчик продал сайт компании Automattic (владеет блог-платформой WordPress).

Молодые люди потратили «несколько сотен долларов» на покупку домена у VPS-провайдера Slicehost, регистрацию компании Logical Awesome и графику для будущего сайта.

Кроме того, разработчики приобрели в фотостоке iStockphoto права на рисунок «осьмикота» (смесь кота и осминога) художника Саймона Оуксли, создавшего первый логотип Twitter. Выдуманное животное стало маскотом GitHub.

Должность генерального директора компании занял Ванстрас. Первое время команда GitHub работала удаленно в кофейнях и из дома, общаясь через сервис Campfire. Собственный офис в Сан-Франциско у компании появился только в 2010 году.

В начале 2008 года разработчики запустили бета-версию платформы и разослали приглашения друзьям и знакомым. Первые участники начали пользоваться GitHub, размещая на площадке свои рабочие коды. Одним из первых проектов, который присоединился к новой платформе, стал фреймворк Ruby on Rails.

В феврале 2008 к команде основателей проекта присоединился Пи Джей Хайетт. С ним в 2007 году Ванстрас познакомился в Cnet, а затем молодые люди пытались запустить аналог Google+ под названием Famspam. Однако их стартап не стал успешным.

Публичный запуск GitHub состоялся 10 апреля 2008 года. К этому моменту проект содержал 20 тысяч репозиториев, а количество пользователей составило более двух тысяч.

«Мы действительно гордимся этим сайтом, и у нас есть так много идей на будущее. Это только начало», — написал в блоге проекта Ванстрас.

Разработчики заложили в GitHub такие возможности, как дублирование или форкинг (Forking), запросы на включение в проект изменений (Pull Requests) и слияние (Merging).

В GitHub можно скопировать любой опубликованный репозиторий в свой профиль, чтобы модифицировать его. Затем разработчик может поделиться изменениями с владельцем репозитория посредством запроса на включение. Если владельцу нравятся изменения, то он может слить их с первоначальным репозиторием.

Таким образом подход GitHub позволил любому зарегистрированному пользователю делиться, улучшать или развивать открытые проекты.

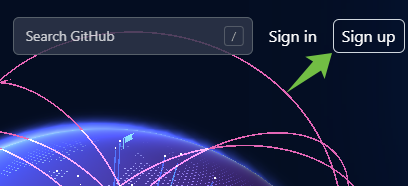
Также GitHub предоставил такие функции совместной работы, как отслеживание ошибок, запрос функций, управление задачами и вики-страницами для каждого проекта.

**Описание.**

1. **Регистрация.**

Официальный сайт web-сервиса: <https://github.com/>

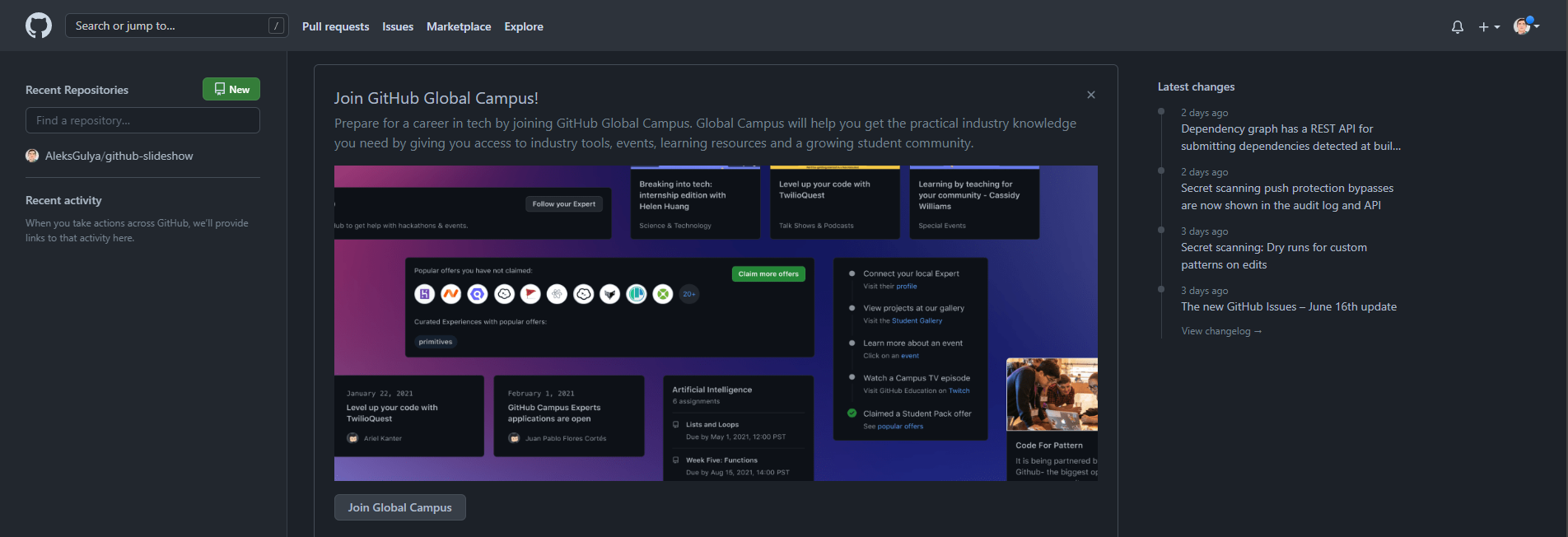
* 1. Проходим регистрацию на сайте по кнопке: Sign up



* 1. Указываем e-mail для регистрации, вводим пароль и создаем аккаунт в репозитории github.

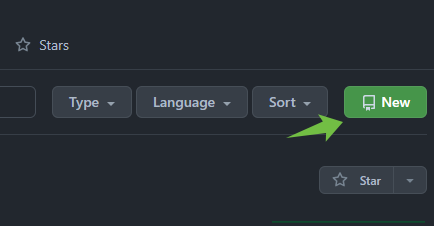


* 1. Далее авторизуемся в сервисе и попадает в главное меню.

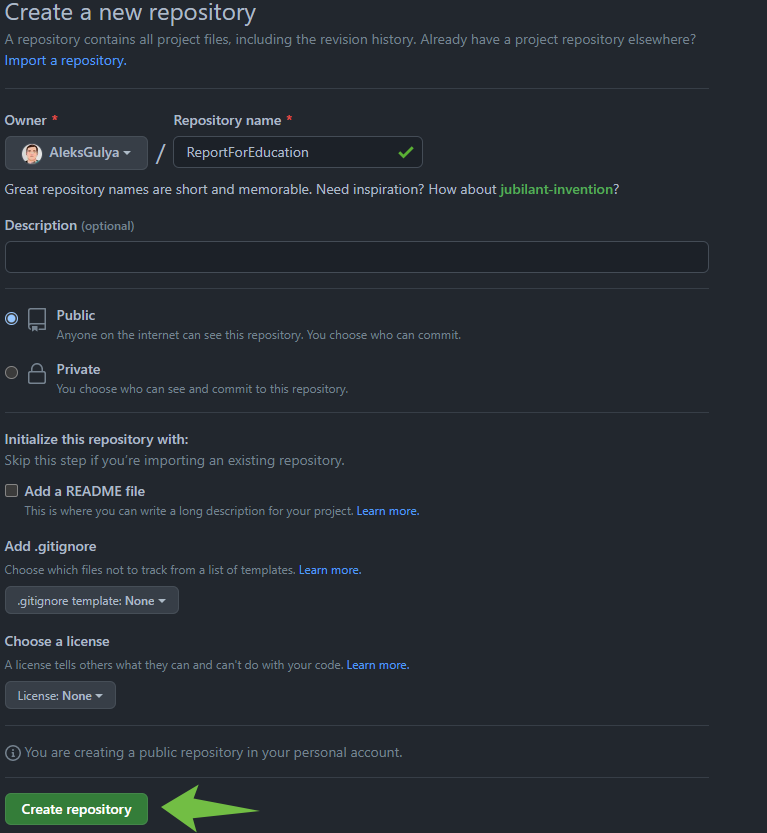


1. **Создание нового репозитория.**
   1. Самый простой способ создания репозитория через интерфейс web-сервиса

Создаем новый репозиторий:



* 1. Указываем имя репозитория и настройки публикации:



1. **Создание репозитория через команды.**

Первоначальная настройка:

* 1. Настройка информации о пользователе для всех локальных репозиториев

$ git config --global user.name "[имя]"

Устанавливает имя, которое будет отображаться в поле автора у выполняемых вами коммитов

$ git config --global user.email "[адрес электронной почты]"

Устанавливает адрес электронной почты, который будет отображаться в информации о выполняемых вами коммитах

Создание репозитория:

* 1. Создание нового репозитория или получение его по существующему URL-адресу

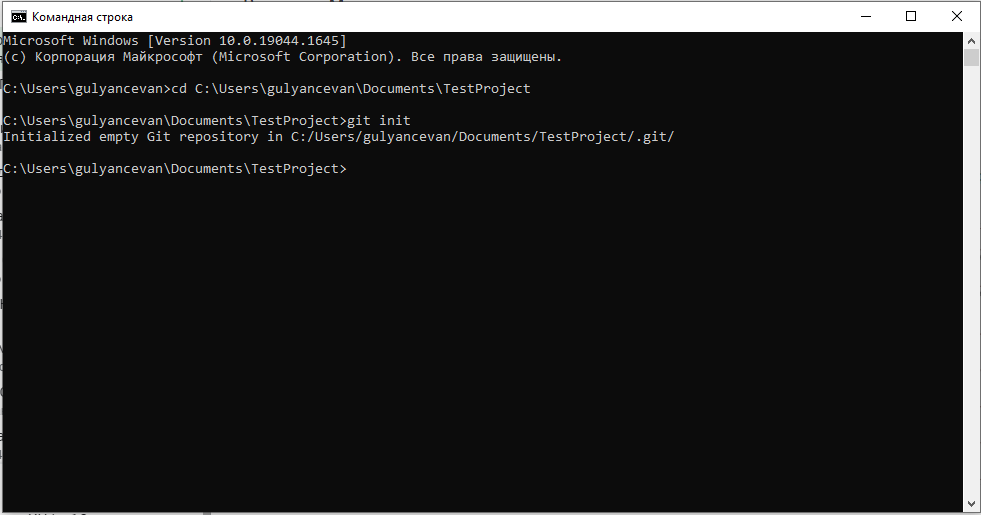
$ git init [название проекта]

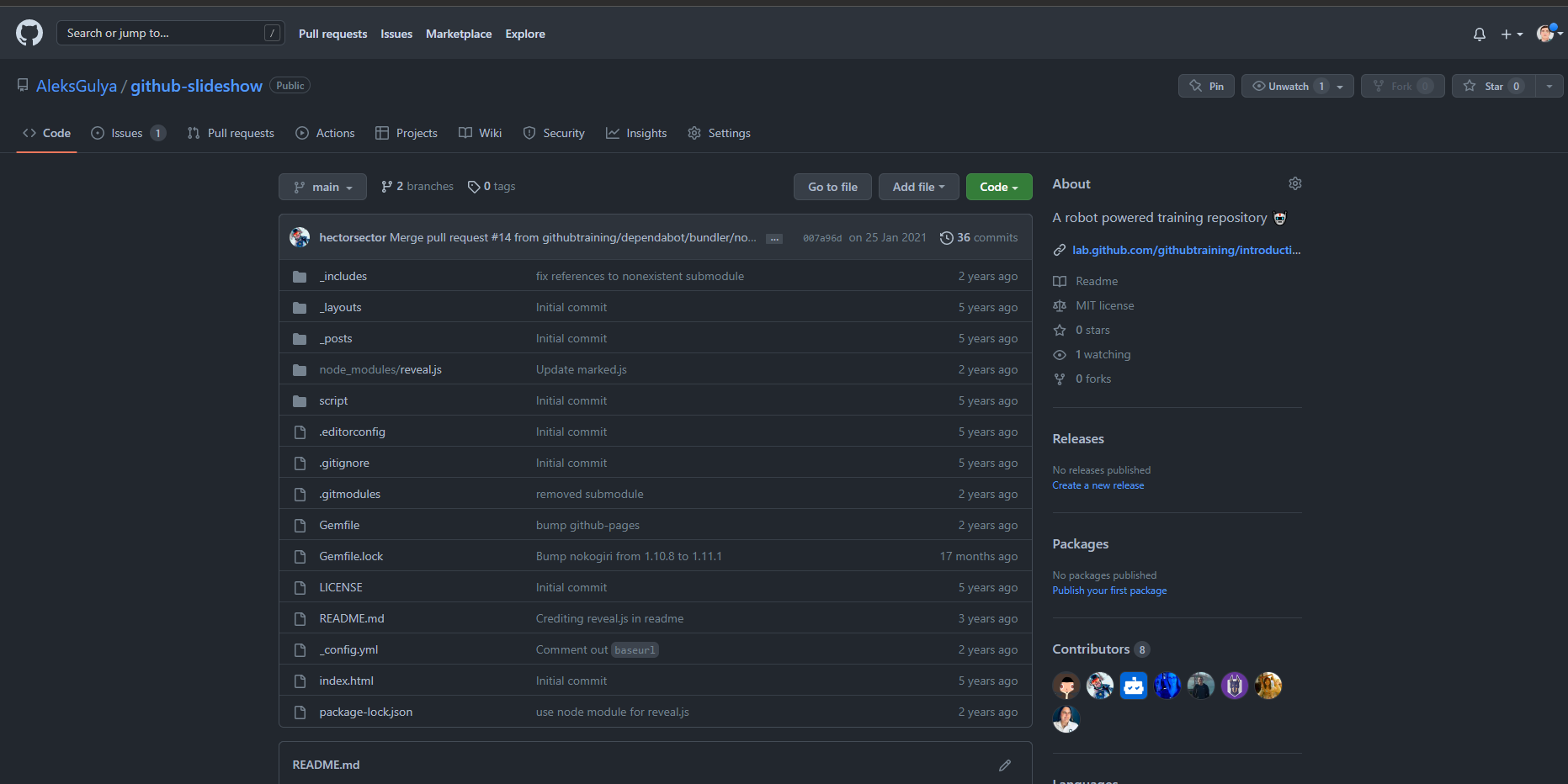
Создаёт новый локальный репозиторий с заданным именем

$ git clone [url-адрес]

Скачивает репозиторий вместе со всей его историей изменений

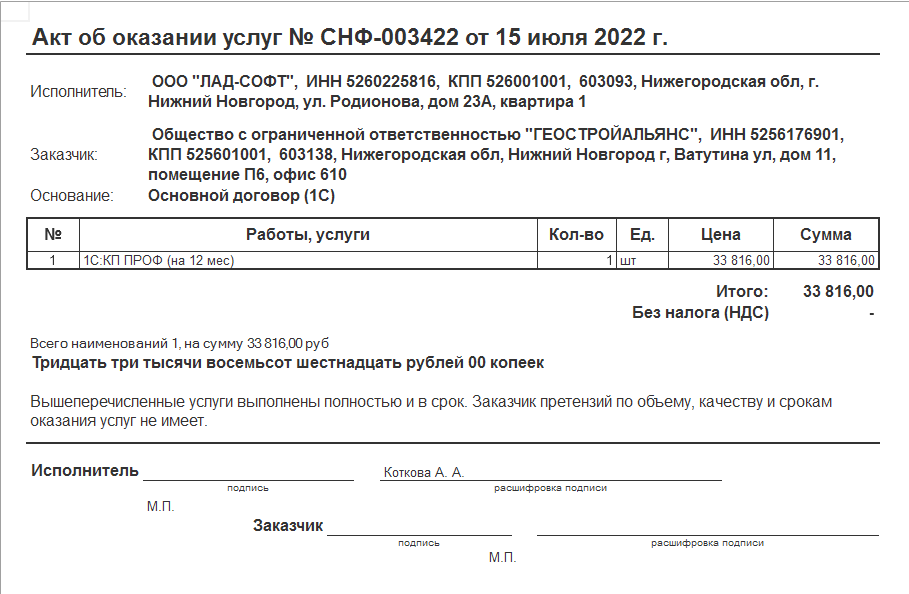
1. **Добавляем существующий репозиторий наш проект на GitHub**





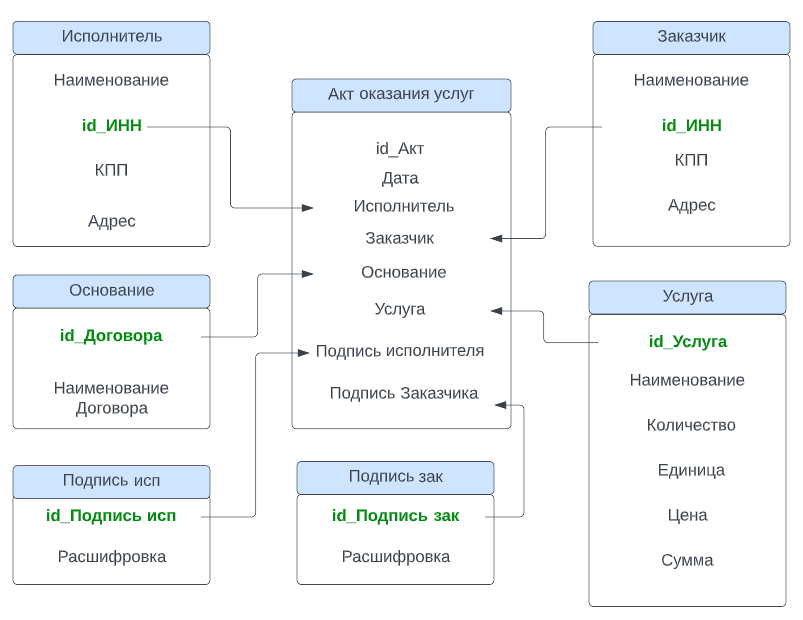
После добавления проекта в репозиторий доступна совместная работа над проектом и вся информация о проекте в web-сервисе.

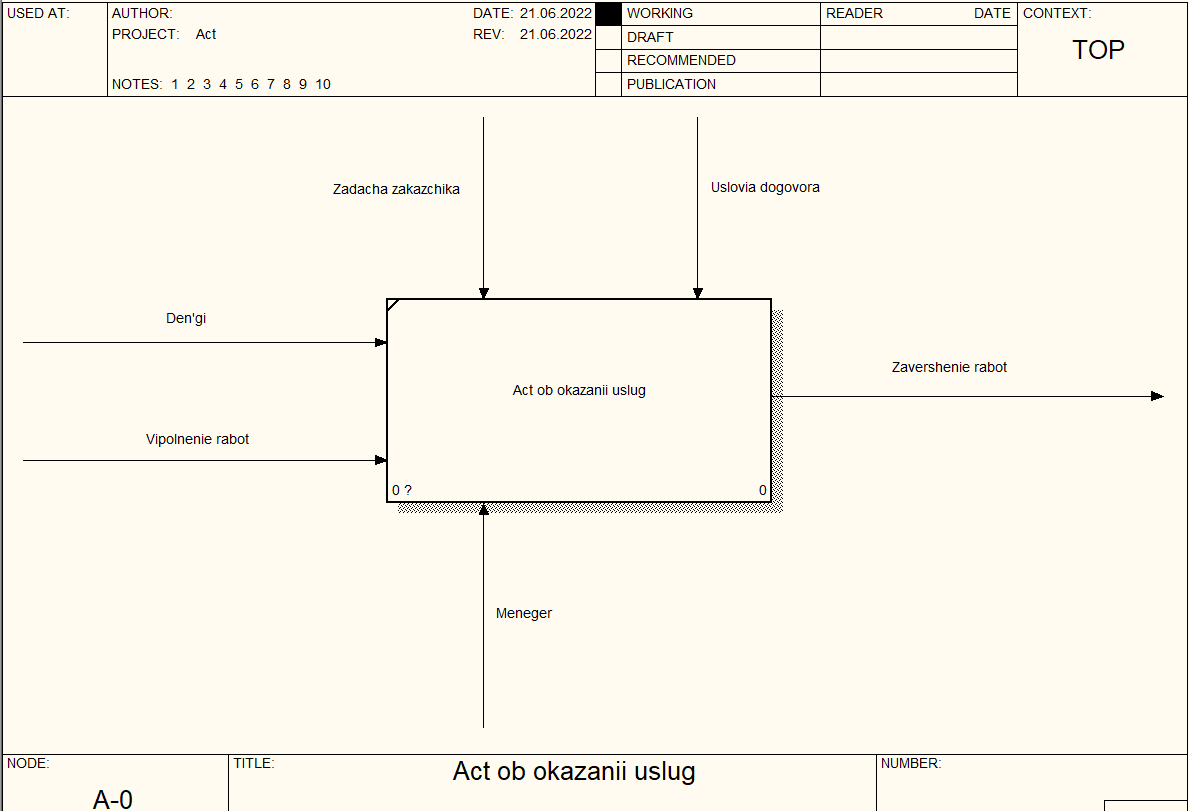
**Проведение реквизитного анализа**



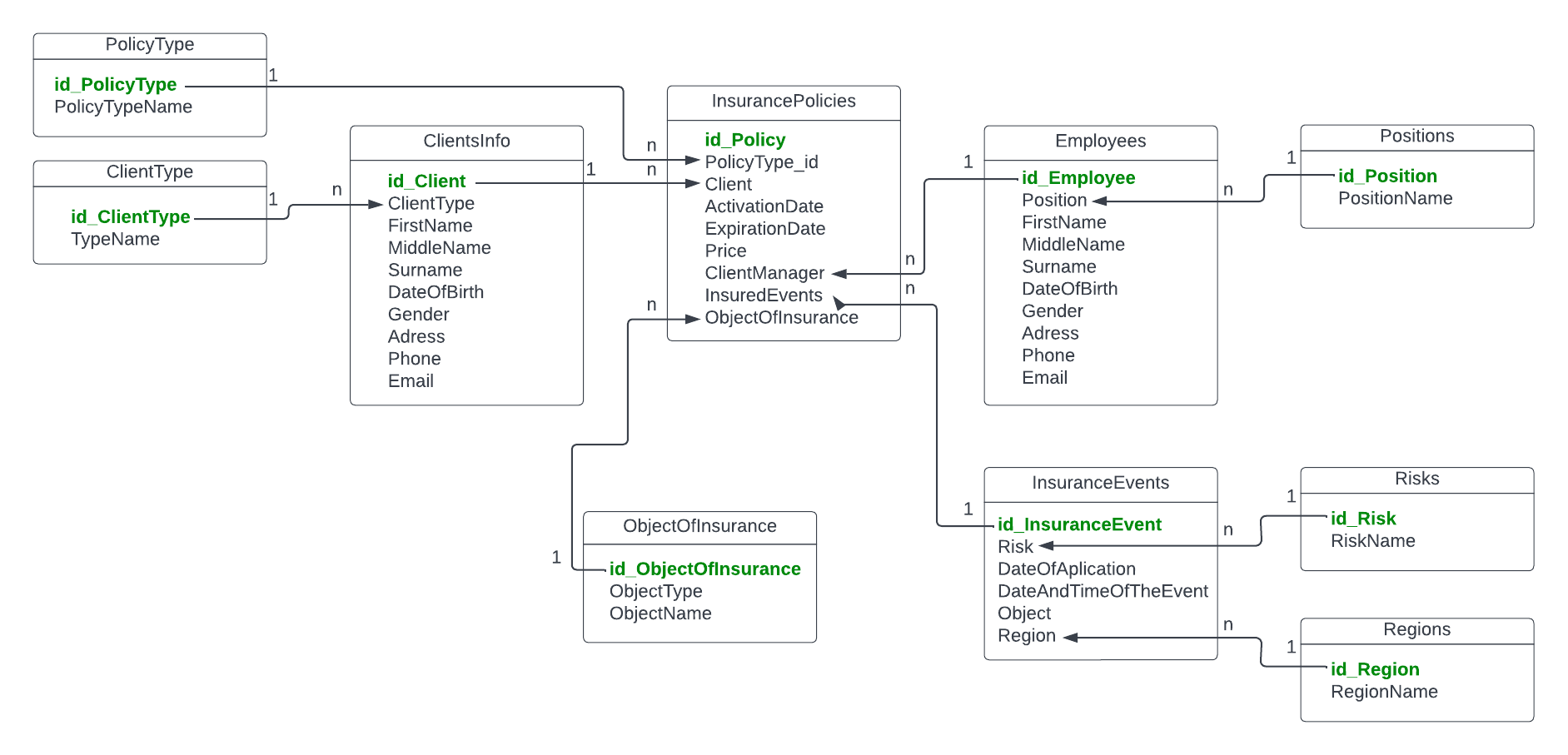
|  |  |
| --- | --- |
| Сущность | Реквизит |
| Акт об оказании услуг | номер |
| дата |
| Исполнитель | Наименование |
| ИНН |
| КПП |
| Адрес |
| Заказчик | Наименование |
| ИНН |
| КПП |
| Адрес |
| Основание | Договор |
| Услуга | номер |
| наименование |
| количество |
| единица |
| цена |
| сумма |
| Подпись исполнителя | подпись |
| расшифровка |
| Подпись заказчика | подпись |
| расшифровка |

Модель реляционной базы данных на основе сущностей и их реквизитов.





**Создание БД по федеральному классификатору по предметной области «Страховая компания» + схема БД. (T-SQL)**

****

**Запрос на создание БД и таблиц**

CREATE TABLE PolicyType (

id\_PolicyType INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

PolicyTypeName VARCHAR(100) NOT NULL

)

GO

CREATE TABLE ClientType (

id\_ClientType INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

TypeName VARCHAR(100) NOT NULL

)

GO

CREATE TABLE ClientsInfo (

id\_Client INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

ClientType INT NOT NULL FOREIGN KEY REFERENCES ClientType (id\_ClientType),

FirstName NVARCHAR(50) NOT NULL,

MiddleName NVARCHAR(50) NOT NULL,

Surname NVARCHAR(50) NOT NULL,

DateOdBirth DATE NOT NULL,

Gender CHAR(1) NOT NULL,

Adres VARCHAR(100) NOT NULL,

Phone VARCHAR(10) NOT NULL UNIQUE,

Email VARCHAR(50) NOT NULL UNIQUE

)

GO

CREATE TABLE ObjectOfInsurance (

id\_ObjectOfInsurance INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

ObjectType VARCHAR(50) NOT NULL,

ObjectName VARCHAR(50) NOT NULL

)

GO

CREATE TABLE Positions (

id\_Position INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

PositionName VARCHAR(50) NOT NULL

)

GO

CREATE TABLE Employees (

id\_Employee INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

Position INT NOT NULL FOREIGN KEY REFERENCES Positions (id\_Position),

FirstName NVARCHAR(50) NOT NULL,

MiddleName NVARCHAR(50) NOT NULL,

Surname NVARCHAR(50) NOT NULL,

DateOdBirth DATE NOT NULL,

Gender CHAR(1) NOT NULL,

Adres VARCHAR(100) NOT NULL,

Phone VARCHAR(10) NOT NULL UNIQUE,

Email VARCHAR(50) NOT NULL UNIQUE

)

GO

CREATE TABLE Risks (

id\_Risk INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

RiskName VARCHAR(100) NOT NULL

)

GO

CREATE TABLE Regions (

id\_Region INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

RegionName VARCHAR(100) NOT NULL

)

GO

CREATE TABLE InsuranceEvents (

id\_InsuranceEvent INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

Risk INT NOT NULL FOREIGN KEY REFERENCES Risks (id\_Risk),

DateOfAplication DATE NOT NULL,

DateAndTimeOfTheEvent DATETIME NOT NULL,

Objct VARCHAR(50) NOT NULL,

Region INT NOT NULL FOREIGN KEY REFERENCES Regions (id\_Region)

)

GO

CREATE TABLE InsurancePolicies (

id\_Policy INT IDENTITY(1,1) NOT NULL PRIMARY KEY,

PolicyType\_id INT NOT NULL FOREIGN KEY REFERENCES PolicyType (id\_PolicyType),

Client INT NOT NULL FOREIGN KEY REFERENCES ClientsInfo (id\_Client),

ActivationDate DATE NOT NULL,

ExpirationDate DATE NOT NULL,

Price MONEY NOT NULL,

ClientManager INT NOT NULL FOREIGN KEY REFERENCES Employees (id\_Employee),

InsuredEvents INT NOT NULL FOREIGN KEY REFERENCES InsuranceEvents (id\_InsuranceEvent),

ObjectOfInsurance INT NOT NULL FOREIGN KEY REFERENCES ObjectOfInsurance (id\_ObjectOfInsurance)

)

GO

**Запрос на вставку данных**

INSERT PolicyType VALUES

('ОСАГО'),

('КАСКО'),

('Несчастный случай'),

('Имущество'),

('Антиклещ')

GO

INSERT ClientType VALUES

('физическое лицо'),

('индивидуальный предприниматель'),

('юридическое лицо'),

('казначейство')

GO

INSERT ClientsInfo VALUES

(1, 'Сергей', 'Георгиевич', 'Смирнов', '20.11.1990', 'М', 'Нижний Новгород, ул. Белинского', '9503335544', 'smirnovsg@gmail.com'),

(3, 'Елена', 'Викторовна', 'Романова', '11.08.1995', 'Ж', 'Москва, ул. Ленина', '9309875544', 'romanovaev@gmail.com'),

(1, 'Никита', 'Алексеевич', 'Коваленко', '13.02.1984', 'М', 'Нижний Новгород, ул. Окская', '9203331010', 'kovalenkona@gmail.com'),

(2, 'Ольга', 'Сергеевна', 'Петрова', '26.12.1993', 'Ж', 'Кстово, ул. Центральная', '9504455526', 'petrovaos@gmail.com'),

(4, 'Александр', 'Владимирович', 'Губернаторов', '15.06.1972', 'М', 'Нижний Новгород, ул. Баумана', '9804545324', 'gubernatorovav@gmail.com')

GO

INSERT ObjectOfInsurance VALUES

('Недвижемость', 'Частный дом'),

('Недвижемость', 'Квартира'),

('Транспортное средство', 'Легковой автомобиль'),

('Транспортное средство', 'Грузовой автомобиль'),

('Человек', 'Взрослый'),

('Человек', 'Ребенок')

GO

INSERT Positions VALUES

('Руководитель'),

('Клиент-менеджер'),

('Старший клиент-менеджер'),

('Ведущий клиент-менеджер')

GO

INSERT Employees VALUES

(1, 'Владимир', 'Викторович', 'Сенаторов', '21.10.1986', 'М', 'Нижний Новгород, ул. Заводская', '9104435142', 'senvv@insurance.com'),

(2, 'Марина', 'Николаевна', 'Титова', '28.12.1988', 'Ж', 'Краснодар, ул. Центральная', '9205555162', 'titmn@insurance.com'),

(3, 'Дмитрий', 'Николаевич', 'Мостовой', '01.01.1993', 'М', 'Нижний Новгород, ул. Деревообделочная', '9088845642', 'mosdn@insurance.com'),

(4, 'Роман', 'Валентинович', 'Филимонов', '22.05.1989', 'М', 'Нижний Новгород, ул. Волжская набережная', '9309605672', 'filrv@insurance.com')

GO

INSERT Risks VALUES

('ДТП'),

('Пожар'),

('Несчастный случай'),

('Происшествие')

GO

INSERT Regions VALUES

('Нижегородская область'),

('Московская область'),

('Питерская область'),

('Республика Татарстан')

GO

INSERT InsuranceEvents VALUES

(1, '21.05.2021', '21.05.2021 22:00', 'Легковой автомоблиь', 1),

(3, '20.03.2022', '20.03.2022 15:00', 'Ребенок', 4),

(2, '05.05.2020', '05.05.2020 23:00', 'Квартира', 1)

GO

INSERT InsurancePolicies VALUES

(1, 3, '31.08.2021', '31.08.2022', '9000', 2, 1, 3),

(4, 2, '10.05.2021', '10.05.2022', '15000', 4, 2, 2)

GO

**Отчет ознакомления с программой Орг Мастер**

Специалистами БИГ-СПб разработана и развивается профессиональная система бизнес-моделирования нового поколения - программа ОРГ-МАСТЕР®.

ОРГ-МАСТЕР решает весь комплекс задач моделирования деятельности компании на структурном и процессном уровнях, а также формирует документированные процедуры - регламенты выполнения бизнес-процессов и организационные Положения. Пакет Должностных инструкций строится в четком соответствии с организационно-функциональной и процессной моделями предприятия.

ОРГ-МАСТЕР существенно дополняет управленческие возможности компании, что повышает ее конкурентоспособность.

При создании бизнес-модели компании формируется база знаний о ее целях, структурах, ресурсах, процессах и других объектах управления. Наличие полной процессной бизнес-модели позволяет оперативно и наглядно получать информацию «как организована деятельность компании» и помогает принимать управленческие решения по ее изменению, создавая и корректируя необходимые регламенты.

Средствами ОРГ-МАСТЕР строится базовая организационно-функциональная модель состоящая из иерархически упорядоченных классификаторов функций-процессов, организационных звеньев, ресурсов и документов, также отношений между ними. Это позволяет провести первичный управленческий анализ организационных аспектов деятельности предприятия.

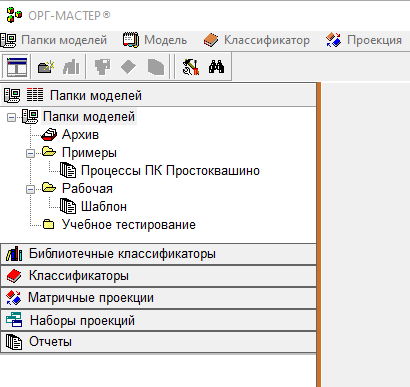
На следующем этапе программа позволяет реализовать полное описание системы процессов компании. Для каждого идентифицированного процесса определяется его целевое назначение, сопряженные процессы: предшественник и потребитель, информационное взаимодействие и обрабатываемые материальные ресурсы, средства реализации и регламенты исполнения.

В результате формируется первичный набор базовых организационных документов: Положения об организационной структуре, Положения о функциональных областях управления и бизнес-направлениях компании, Положения о подразделениях (службах, отделах), Должностных инструкций, а также спецификации выделенных процессов компании

Таким образом, программа обеспечивает реализацию начального этапа постановки регулярного менеджмента в компании.

**Использование программы (функциональность)**

В разделе папки моделей, мы можем посмотреть пример процессов ПК Простоквашино, а также рассмотреть предлагаемый Шаблон



Ниже можно увидеть все классификаторы данной модели

Создание нового классификатора по образцу из модели

открытие

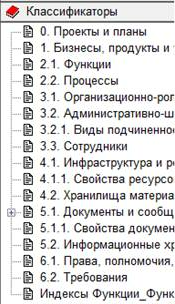
изменение свойств

сохранение как библиотечный классификатор в директорию

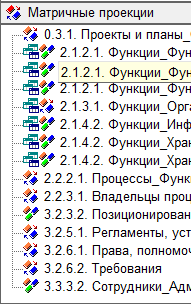
очищение

восстановление

удаление

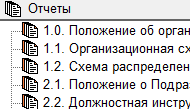


Матричные проекции - содержит перечень матричных проекций открытой в данный момент модели.

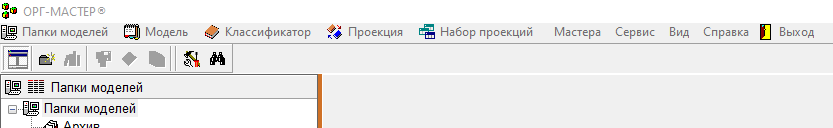


Наборы проекций - содержит перечень наборов проекций открытой в данный момент модели.

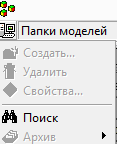
Отчеты - содержит перечень отчетов открытой в данный момент модели. бизнес моделирование матричный отчет



Соответственно сверху находятся Разделы программы ОРГ-МАСТЕР

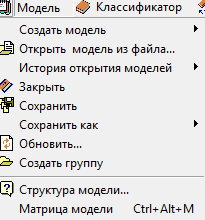


В разделе папки моделей, возможно создание, свойства, удаление модели, а также есть поиск уже существующих и архив.



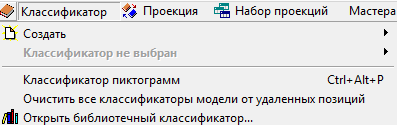
Модель - опции работы с моделями (создание, изменение свойств, сохранение, удаление).

В разделе модель содержатся следующие подразделы:



Классификатор - частичное дублирование опций контекстного меню элементов классификатора и меню открытого классификатора, а так же опции работы с содержимым классификатора (сортировка, сохранение, импорт).

В классификаторе содержатся следующие подразделы:



Мастера - запуск мастеров оргдиаграмм и диаграмм бизнес-процессов.

Сервис - вспомогательные опции, не используемые в процессе обучения.

Вид - настройки параметров отображения модели.

Справка - вызов справочной информации о продукте, методологии и компании.

В разделе справка можно узнать подробную информацию о программе, вызвать справку или написать в службу поддержки.

Выход - выход из программы

Создание модели:

