

УДК 004.91

**ФУНКЦИОНАЛЬНОЕ ПРОЕКТИРОВАНИЕ ИНФОРМАЦИОННОЙ
СИСТЕМЫ ОПТИМИЗАЦИИ РАБОТЫ СОТРУДНИКОВ
ЮРИДИЧЕСКОЙ ФИРМЫ****Иванов Сергей Александрович,**

кандидат технических наук, доцент,

Санкт-Петербургский государственный лесотехнический университет им. С.М. Кирова

Кочерженко Андрей Александрович,

студент 4-го курса обучения,

Санкт-Петербургский государственный лесотехнический университет им. С.М. Кирова

Иванов Максим Александрович,

Студент 2-го курса обучения,

Астраханский Государственный Технический Университет

Аннотация

В современном мире информационные технологии играют ключевую роль в повышении эффективности и оптимизации рабочих процессов. Одним из перспективных направлений является разработка специализированных программных решений, направленных на автоматизацию рутинных операций и оптимизацию рабочего процесса. В рамках данной статьи рассматривается разработка телеграм-бота, который позволит оптимизировать рабочие процессы, сократить время на выполнение рутинных задач и повысить эффективность работы сотрудников.

Ключевые слова: диаграмма деятельности, функциональное моделирование, нотация IDEF0.

**FUNCTIONAL DESIGN OF AN INFORMATION SYSTEM FOR OPTIMIZING
THE WORK OF LAW FIRM EMPLOYEES****Ivanov Sergey Alexandrovich,**

Candidate of Technical Sciences, Associate Professor,

Saint Petersburg State Forest Technical University named after S.M. Kirova

kemsit@mail.ru

Kocherzhenko Andrey Alexandrovich,

4th year student,

Saint Petersburg State Forest Technical University named after S.M. Kirova

Ivanov Maksim Aleksandrovich,

2th year student,

Astrakhan State Technical University
 Maksfire2001@mail.ru

ABSTRACT

In the modern world, information technology plays a key role in improving efficiency and optimizing work processes. One of the promising areas is the development of specialized software solutions aimed at automating routine operations and optimizing the workflow. This article discusses the development of a telegram bot that will optimize work processes, reduce time for routine tasks and increase employee efficiency.

Keywords: activity diagram, functional modeling, IDEF0 notation.

Предварительный анализ процессов проводится с помощью описания процессов по нотации IDEF0 [1,2] (рисунок 1), где отражено текущее состояние процессов обработки POST-запроса [3,4].

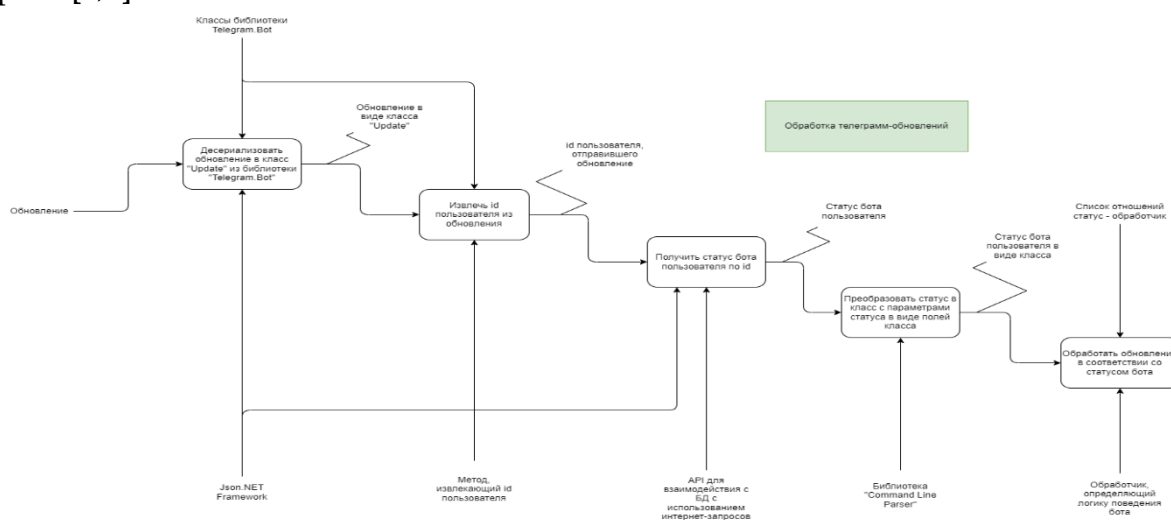


Рисунок 1 – «описание процессов в нотации IDEF0»

При получении телеграмм-обновления в формате POST-запроса начинается процесс приведения тела POST-запроса к классу «Update» из библиотеки «Telegram.Bot».

Далее из телеграмм-обновления в виде класса «Update» извлекается id пользователя, от лица которого было отправлено обновление. В силу особенностей изменения структуры телеграмм-обновления, был выделен отдельный программный модуль, в котором были учтены все возможные случаи структуры телеграмм-обновления в рамках функционала телеграмм-бота. Подобный программный модуль позволяет извлекать id пользователя без учета возможных изменений в структуре телеграмм-обновления.

С помощью извлеченного ранее id пользователя телеграмм-бот обращается к «API для взаимодействия с базой данных» для получения актуального статуса бота конкретного пользователя.

Полученный статус бота представляет собой строку с определёнными параметрами. Для упрощения работы, статус бота преобразовывается ранее определённому классу, где в поля класса заносятся значения параметров статуса. Механизм преобразования осуществляется с помощью библиотеки «Command Line Parser».

Последним шагом результаты преобразования статуса сопоставляются со списком отношений «статус – обработчик». Если был найден обработчик, соответствующий статусу

бота, то работа с обновлением передаётся найденному обработчику, в противном случае – телеграмм-бот никак не реагирует на отправленное обновление.

В модель не была добавлена отправка ответа на отправленный POST-запрос со стороны телеграмма из-за того, что на данный момент в качестве ответа будет возвращаться «ОК» с кодом «200», говорящий об успешной обработке телеграмм-обновления.

В дополнение к предварительному описанию процессов по нотации IDEF0 была составлена диаграмма деятельности [5], на которой отражены возможные ветвления процессов (Рисунок 2).

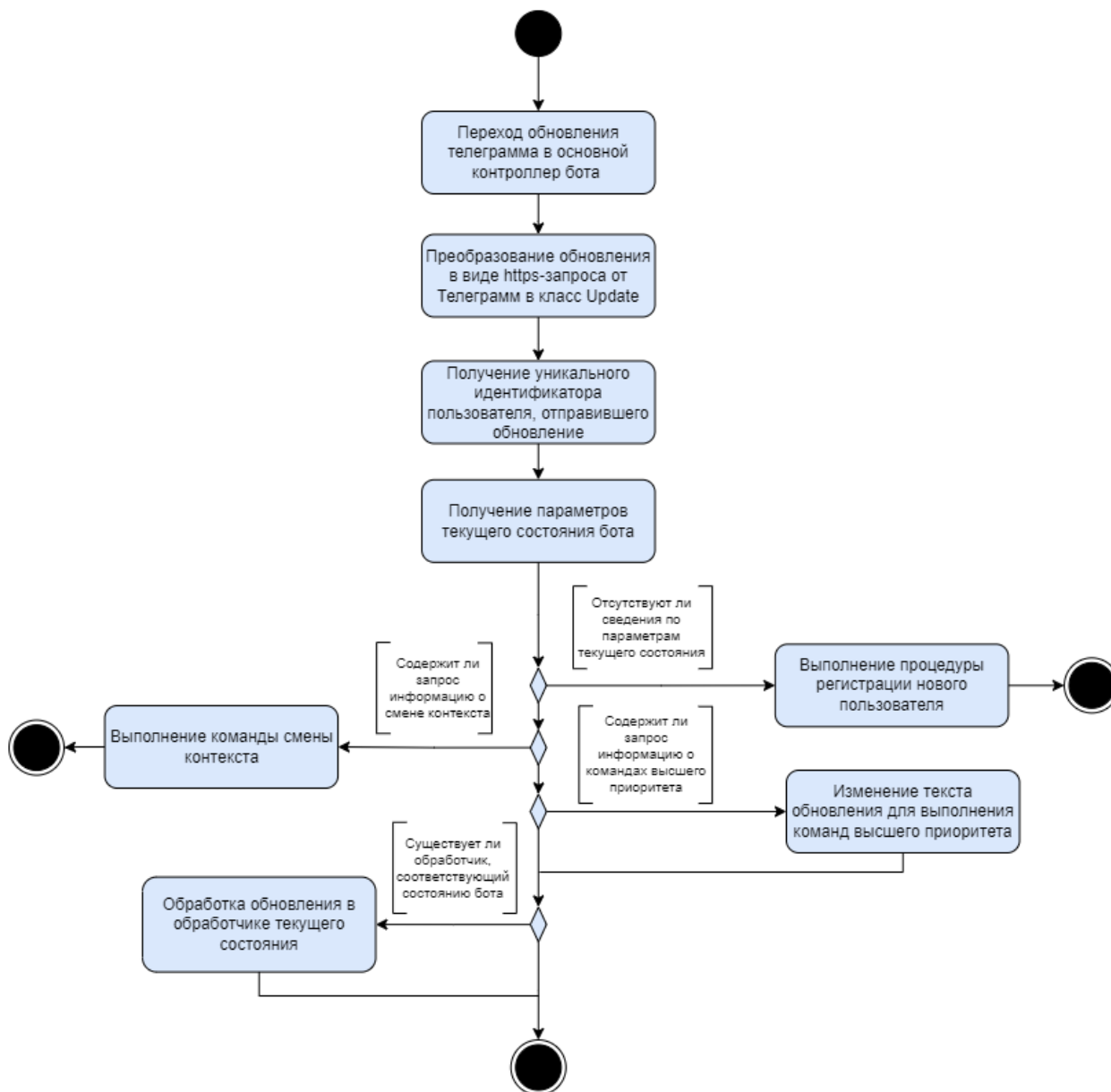


Рисунок 2 – «Диаграмма деятельности в дополнение к предварительному анализу процессов в нотации IDEF0»

На диаграмме деятельности показаны возможные ветвления обработки POST-запроса от мессенджера «Телеграмм»:

В случае отсутствия информации по статусу бота конкретного пользователя, бот начинает процедуру регистрации нового пользователя.

В случае наличия информации о командах смены контекста, обработка POST-запроса заканчивается на выполнении команды смены контекста.

В случае наличия информации о командах высшего приоритета, меняется текстовое значение телеграмм-обновления. Это действие связано с особенностями работы отдельных обработчиков из вышеупомянутого списка отношений «статус – обработчик».

В случае существования обработчика, соответствующему статусу бота, работа с телеграмм-обновлением передаётся найденному обработчику, иначе бот никак не реагирует на отправленное телеграмм-обновление.

Проектная документация телеграм-бота составлена в нотации IDEF0 (Рисунок 3), отражающей состояние системы после внедрения изменений.

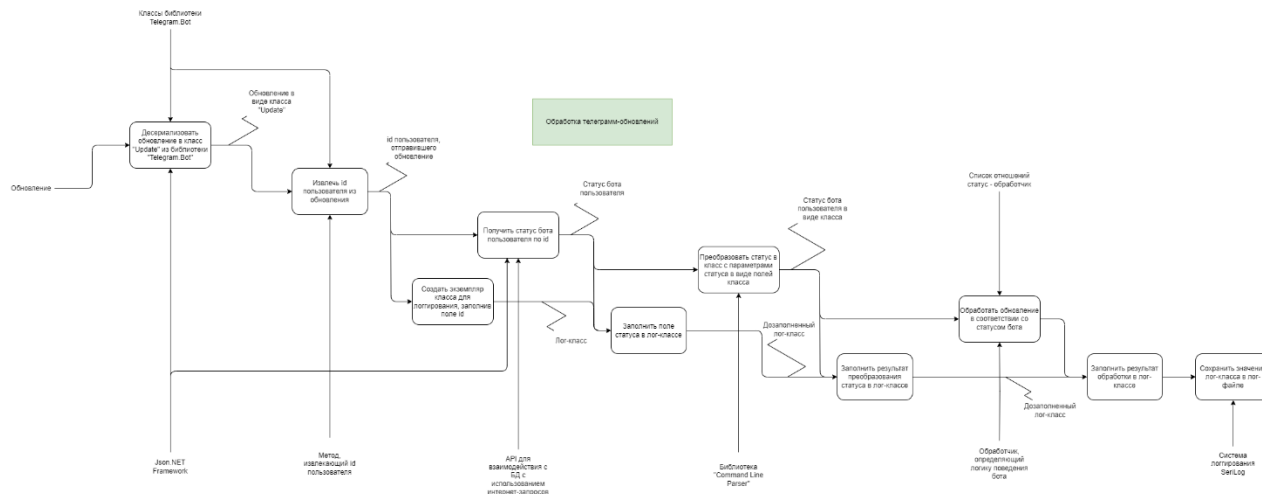


Рисунок 3 – «Проектная документация телеграм-бота в нотации IDEF0»

В соответствии с документацией проекта, должны быть добавлены процессы логирования некоторых элементов обработки телеграмм-обновлений.

Процессы логирования призваны упростят процесс запуска телеграмм-бота на удалённом сервере, где часто нет возможности отладить программу во время её работы. Внедрение логирования также позволит собирать аналитику на основе данных, записываемых в лог-файлы текстового формата.

На диаграмме деятельности (Рисунок 4) были добавлены операции логирования, таким образом количество конечных узлов сократилось до одного из-за необходимости логирования всех сценариев поведения телеграмм-бота.

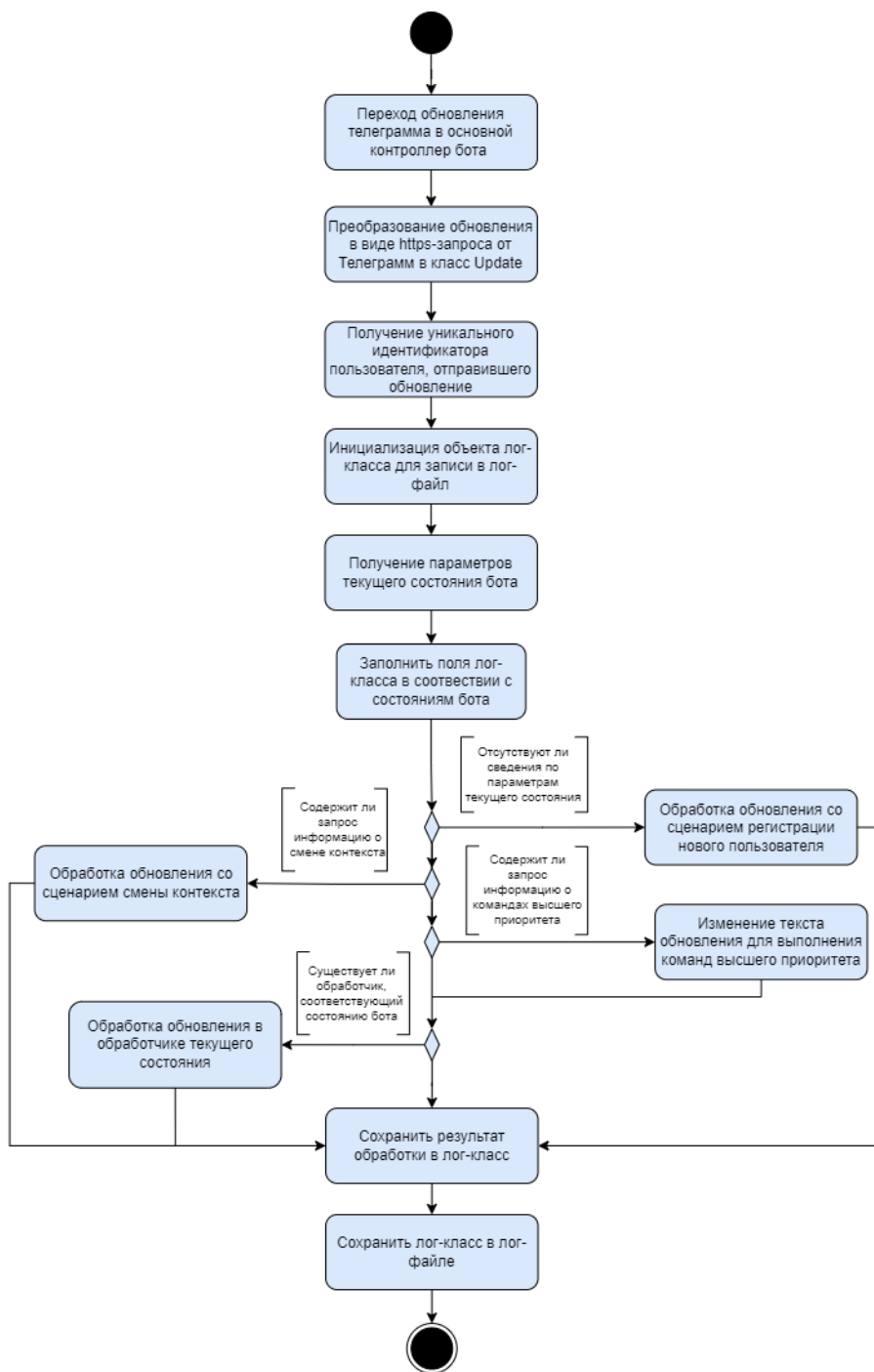


Рисунок 4 – «Диаграмма деятельности к разработанной документации проекта в нотации IDEF0»

Архитектура предлагаемого решения выглядит следующим образом (Рисунок 5):

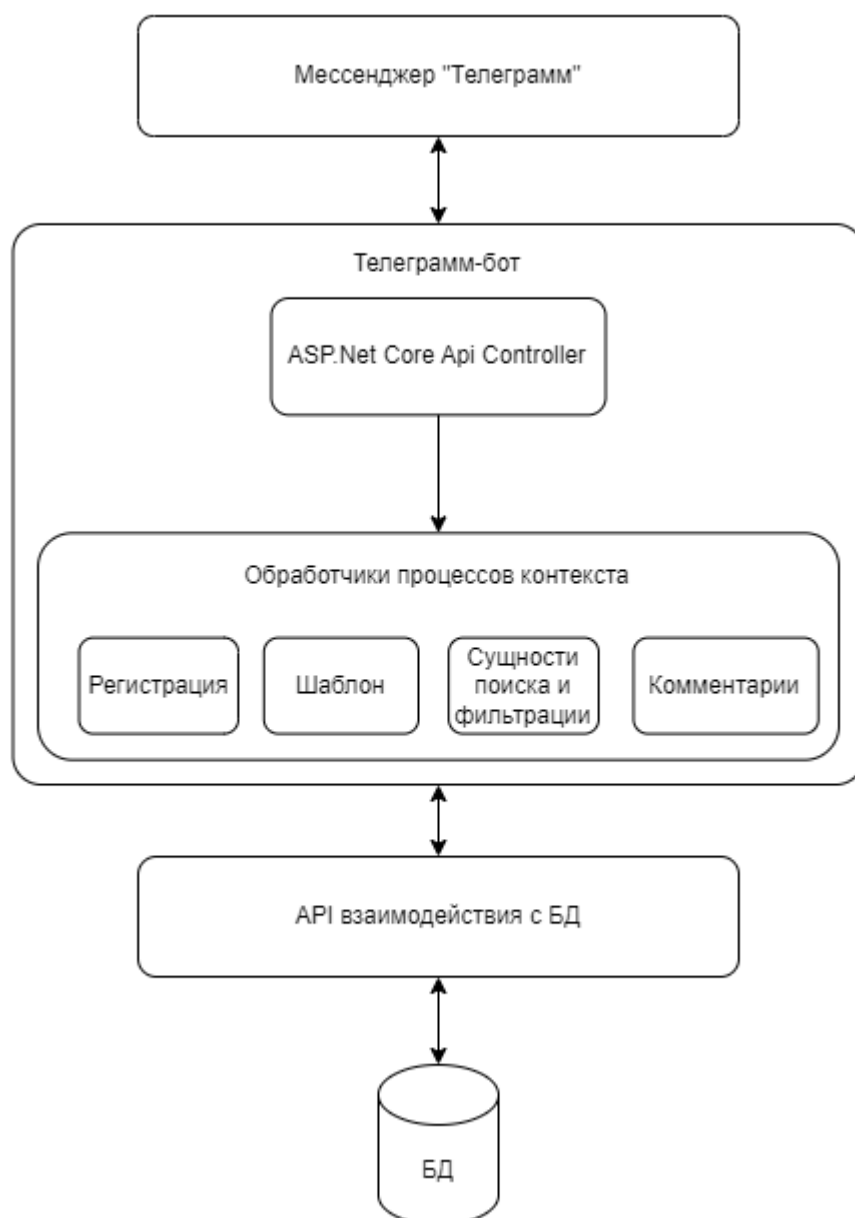


Рисунок 5 –«Архитектура предлагаемого решения»

Предлагаемое решение представляет собой клиент-серверное приложение [6], состоящее из четырёх основных компонентов:

Клиент. В качестве клиентской части выступает мессенджер «Телеграмм», который инициирует отправку POST-запроса серверу, где размещён телеграмм-бот

Сервер, телеграмм-бот, который принимает POST-запрос от клиента, обрабатывает его и определяет, как на него реагировать

Сервер, содержащий в себе API-эндпоинты для взаимодействия с базой данных

База данных, которая хранит в себе информацию, на основе которой ведётся поиск сущностей. В эту же базу данных добавляются новые данные.

В ходе проектирования программного решения был проведён предварительный анализ процессов и разработана документация проекта в соответствии с нотацией IDEF0, к которым прилагаются диаграммы активности для показа ветвлений программы, предварительный анализ процессов позволил выявить недостатки и проблемы посредством описания текущего состояния процессов в программе. В результате анализа модели было решено добавить систему логгирования параметров выполнения программы в файл.

Разработанная документация проекта отражает процессы после добавления системы логирования, что позволяет оценить временные затраты добавления системы в программный код.

Кроме того, была разработана архитектура предлагаемого решения, наглядно отображающая общий вид системы и основные программные компоненты.

Список литературы:

1. Перспективы развития цифрового общества в современных условиях / С. В. Удахина, А. С. Ушакова, Д. Е. Федорова [и др.]. – Санкт-Петербург : Санкт-Петербургский университет технологий управления и экономики, 2019. – 78 с.
2. Таюрская, И. С. Проектирование информационных систем : Учебно-методическое пособие для подготовки студентов, обучающихся по направлению 09.03.03 Прикладная информатика / И. С. Таюрская. – Санкт-Петербург : Санкт-Петербургский университет технологий управления и экономики, 2022. – 153 с.
3. Михаил Фокеев. Проектирование процессов // vc.ru. 2024 [Электронный ресурс]. URL: <https://vc.ru/life/948641-proektirovanie-processov>
4. Пособие по HTTP-запросам в Python и Web API [Электронный ресурс]. URL: <https://pythonist.ru/posobie-po-http-zaprosam-v-python-i-web-api/>
5. Джейд Моралес. Диаграмма активности UML // mindonmap.com. 2023 [Электронный ресурс]. URL: <https://www.mindonmap.com/ru/blog/uml-activity-diagram/>
6. Разработка клиент-серверных приложений [Электронный ресурс]. URL: <https://sky.pro/wiki/javascript/razrabotka-klient-servernyh-prilozhenij/>

References:

1. Prospects for the development of a digital society in modern conditions / S. V. Udakhina, A. S. Ushakova, D. E. Fedorova [et al.]. – St. Petersburg : St. Petersburg University of Management Technologies and Economics, 2019. – 78 p.
2. Tayurskaya, I. S. Designing information systems : An educational and methodological guide for the preparation of students studying in the direction of 09.03.03 Applied Informatics / I. S. Tayurskaya. – St. Petersburg : St. Petersburg University of Management Technologies and Economics, 2022. – 153 p.
3. Mikhail Fokeev. Process design // vc.ru . 2024 [Electronic resource]. URL: <https://vc.ru/life/948641-proektirovanie-processov>
4. A guide to HTTP requests in Python and the Web API [Electronic resource]. URL: <https://pythonist.ru/posobie-po-http-zaprosam-v-python-i-web-api/>
5. Jade Morales. UML Activity Diagram // mindonmap.com . 2023 [Electronic resource]. URL: <https://www.mindonmap.com/ru/blog/uml-activity-diagram/>
6. Development of client-server applications [Electronic resource]. URL: <https://sky.pro/wiki/javascript/razrabotka-klient-servernyh-prilozhenij/>