

**ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»**

Факультет компьютерных наук
Образовательная программа «Программная инженерия»

СОГЛАСОВАНО

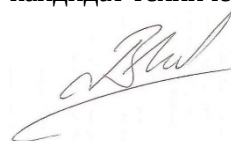
Технический руководитель
внештатный преподаватель департамента
программной инженерии



Н. И. Веселко
«10» апреля 2023 г.

УТВЕРЖДАЮ

Академический руководитель образовательной
программы «Программная инженерия»,
кандидат технических наук



В. В. Шилов
«12» апреля 2023 г.

**МОДУЛЬ УТИЛИТАРНЫХ АБСТРАКЦИЙ МОДЕЛЕЙ ДЛЯ ФРЕЙМВОРКА
DJANGO**

Техническое задание

ЛИСТ УТВЕРЖДЕНИЯ

RU.17701729.04.03-01 ТЗ 01-1-ЛУ

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

Исполнитель
студент группы БПИ203



Н. С. Череминин
«10» апреля 2023 г.

Москва 2023

УТВЕРЖДЕН
RU.17701729.04.03-01 ТЗ 01-1-ЛУ

**МОДУЛЬ УТИЛИТАРНЫХ АБСТРАКЦИЙ МОДЕЛЕЙ ДЛЯ ФРЕЙМВОРКА
DJANGO**

Техническое задание

RU.17701729.04.03-01 ТЗ 01-1

Листов 19

Подп. и дата	
Инв. № дубл.	
Взам. инв. №	
Подп. и дата	
Инв. № подл	

Москва 2023

СОДЕРЖАНИЕ

ГЛОССАРИЙ.....	4
1. ВВЕДЕНИЕ.....	5
1.1. Наименование программы.....	5
1.2. Краткая характеристика и область применения программы.....	5
2. ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ.....	6
2.1. Документы, на основании которых ведется разработка.....	6
2.2. Наименование темы разработки.....	6
3. НАЗНАЧЕНИЕ РАЗРАБОТКИ.....	7
3.1. Функциональное назначение.....	7
3.2. Эксплуатационное назначение.....	7
4. ТРЕБОВАНИЯ К ПРОГРАММЕ.....	8
4.1. Требования к функциональным характеристикам.....	8
4.1.1. Требования к модели логов (LogModel).....	8
4.1.2. Требования к модели логов с началом и окончанием (UpdateLogModel).....	8
4.1.3. Требования к обновляемой модели (UpdatableModel).....	8
4.1.4. Требования к исторической модели (HistoryModel).....	10
4.2. Требования к временным характеристикам.....	10
4.3. Требования к интерфейсу.....	10
4.4. Требования к надежности.....	11
4.5. Условия эксплуатации.....	11
4.6. Требования к составу и параметрам технических средств.....	11
4.7. Требования к информационной и программной совместимости.....	11
4.8. Требования к маркировке и упаковке.....	11
4.9. Требования к транспортировке и хранению.....	11
5. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ.....	12
5.1. Состав программной документации.....	12
5.2. Специальные требования к программной документации.....	12
6. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ.....	13
6.1. Ориентировочная экономическая эффективность.....	13
6.2. Предполагаемая потребность.....	13
6.3. Преимущества разработки по сравнению с аналогами.....	13

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

7. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ.....	14
8. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ.....	16
8.1. Виды испытаний.....	16
8.2. Общие требования к приемке работы.....	16
СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ.....	17
ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ.....	19

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

ГЛОССАРИЙ

- ПО — программное обеспечение [17].
- БД — база данных [19].
- ООП — объектно-ориентированное программирование [20].
- SDK (software development kit) — набор (пакет) средств (инструментов) разработки ПО [15].
- ORM (object-relational mapping) — технология программирования, которая связывает базы данных с концепциями объектно-ориентированных языков программирования, создавая «виртуальную объектную базу данных» [18].
- Модель (данных) — абстрактное, логическое определение объектов, операторов и прочих элементов, в совокупности составляющих абстрактную машину доступа к данным, с которой взаимодействует пользователь [16]. Здесь: описание набора полей, из которых состоит таблица данных, и их свойств, а также свойств самой таблицы, используемое Django ORM [14].
- Кверисет (здесь) — класс, использующийся для получения записей моделей и наследующий `django.db.models.QuerySet`.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

1. ВВЕДЕНИЕ

1.1. Наименование программы

Наименование программы — «Модуль утилитарных абстракций моделей для фреймворка Django» [14].

Наименование программы на английском языке — «Utility Model Abstractions Module for Django Framework» [14].

Краткое наименование программы — «django-abstract-models».

1.2. Краткая характеристика и область применения программы

Программа является набором (пакетом) средств разработки (SDK) для использования на уровне моделей в проектах, написанных с использованием фреймворка Django [14] на языке программирования Python [21]. Этот SDK должен включать ряд абстрактных моделей, облегчающих написание кода и упрощающих создание пользовательских моделей путем наследования данных абстракций. Абстрактные модели должны обладать достаточной гибкостью для пользовательской настройки и иметь переопределяемые поля и методы.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

2. ОСНОВАНИЯ ДЛЯ РАЗРАБОТКИ

2.1. Документы, на основании которых ведется разработка

Разработка ведется в соответствии с учебным планом подготовки бакалавров по направлению 09.03.04 «Программная инженерия» и утвержденной академическим руководителем образовательной программы темы курсового проекта.

2.2. Наименование темы разработки

Наименование темы разработки — «Модуль утилитарных абстракций моделей для фреймворка Django» [14].

Разработка выполняется в рамках темы проектной работы «Модуль утилитарных абстракций моделей для фреймворка Django» [14].

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

3. НАЗНАЧЕНИЕ РАЗРАБОТКИ

3.1. Функциональное назначение

Пакет должен содержать абстрактные модели — Python [21] классы, унаследованные (напрямую или через других родителей) от `django.db.models.Model`. Абстрактные модели должны предоставлять базовый функционал (переопределяемый наследниками) для моделей логов, моделей логов обновления, обновляемых логируемых моделей, а также исторических моделей.

3.2. Эксплуатационное назначение

Пакет предназначен для наследования пользовательскими моделями предоставляемых пакетом абстрактных моделей для упрощения написания кода и единого базового поведения всех наследуемых моделей.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4. ТРЕБОВАНИЯ К ПРОГРАММЕ

4.1. Требования к функциональным характеристикам

Все модели должны быть абстрактными (устанавливается параметр `abstract = True` во вложенном классе `Meta`).

4.1.1. Требования к модели логов (`LogModel`)

Модель должна иметь атрибуты: время записи (тип: `django.db.models.DateTimeField`) (далее: X) и сообщения (тип: `django.db.models.TextField`) (далее: Y).

Модель должна иметь переопределяемые поля: для настройки максимального количества записей (тип: `int`, по умолчанию: `None`) (далее: A).

Модель должна иметь публичную функцию удаления старых логов. Функция должна срабатывать автоматически при создании новой записи любым из средств Django [14] ORM. При значении `None` в поле A функция ничего не делает. Иначе функция проверяет, не превысило ли количество записей значения в поле A, и, если это так, удаляет столько записей наиболее старых записей (по атрибуту X), какова разница между текущим их количеством и значением в поле A.

4.1.2. Требования к модели логов обновления (`UpdateLogModel`)

Модель предполагается использовать в первую очередь для логирования обновлений обновляемых моделей.

Модель должна быть унаследована от модели логов.

Модель должна иметь дополнительные атрибуты: время окончания события (тип: `django.db.models.DateTimeField`) (далее: Z), флаг отмены обновления (тип: `django.db.models.BooleanField`) (далее: K), флаг прерывания обновления (тип: `django.db.models.BooleanField`) (далее: L), флаг ошибки обновления (тип: `django.db.models.BooleanField`) (далее: M).

4.1.3. Требования к обновляемой логируемой модели (`UpdatableLoggableModel`)

Модель должна иметь переопределяемые поля: для настройки логирования обновлений (тип: `Type[UpdateLogModel]`, по умолчанию: `None`) (далее: C), для настройки максимального времени обновления (тип: `datetime.timedelta`, по умолчанию: `None`) (далее: D), для настройки коэффициента

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

допустимого превышения времени обновления (тип: float, по умолчанию: 1) (далее: E), для настройки вычисления ожидаемого времени обновления на основе среднего значения по имеющимся в модели записям (флаг) (тип: bool, по умолчанию: False) (далее: F).

Модель должна при помощи объектов блокировки (threading.Lock) гарантировать, что в любой момент времени запущено не более одного процесса обновления.

Если поле C имеет значение None, то при вызове функций модели может быть выброшено исключение, иначе исключение никогда не должно выбрасываться (за исключением случаев неправильно настройки параметров в моделях-наследниках).

Модель должна иметь публичную функцию для возврата последнего времени обновления.

Модель должна иметь публичную функцию для проверки того, обновляется ли она в настоящий момент. Функция возвращает истину, если объект блокировки занят, иначе — ложь.

Модель должна иметь публичную функцию вычисления среднего времени обновления. Функция возвращает среднее время, вычисленное как разность между полями X и Z модели логов из атрибута C, исключая записи, в которых пусто одно из полей X или Z или любое из полей K, L, M имеет значение истина.

Модель должна иметь публичную функцию проверки того, может ли быть начато новое обновление. Функция возвращает истину, если в данный момент не запущено обновление, или если значение атрибута F — истина и текущее время обновления превышает среднее время обновления, умноженное на значение атрибута E, или если значение атрибута F — ложь (или не удалось вычислить среднее время обновления) и текущее время обновления превышает значение атрибута D. Иначе — возвращает ложь.

Модель должна иметь защищенную функцию обновления. Данная функция должна переопределяться в наследниках. В абстрактной модели функция ничего не делает.

Модель должна иметь публичную функцию обновления. Функция создает запись лога со текущим временем в поле X и запускает публичную функцию проверки того, может ли быть начато обновление. Если та возвращает ложь, функция записывает в лог истину в поле K и текущее время в поле Z и завершается (механизм защиты от множественного перезапуска обновления). Иначе функция оповещает предыдущее обновление о необходимости прерваться, ожидает его прерывания (если предыдущее обновление есть), вызывает защищенную функцию обновления в try-эксперт блоке

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

и при возникновении исключения записывает его в сообщения лога в поле Y, затем при любых обстоятельствах записывает текущее время в поле Z.

4.1.4. Требования к исторической модели (HistoryModel)

Модель должна иметь атрибуты: время записи (тип: `django.db.models.DateTimeField`) (далее: W).

Модель должна иметь переопределяемые поля: для настройки группировки (названия полей) при поиске максимальной даты (тип: `tuple[str, ...] | list[str] | str`, по умолчанию: None) (далее: G), для настройки поля (название этого поля) для поиска максимальной даты (тип: `str`, по умолчанию: None) (далее: H).

Для модели должен быть создан свой кверисет (класс, унаследованный от `django.db.models.QuerySet`) и менеджер (класс, создаваемый по образцу `django.db.models.Manager`).

Созданный кверисет должен иметь публичную функцию, принимающую на вход названия полей для группировки (тип: `tuple[str, ...] | list[str] | str`), название поля даты (тип: `str`) и саму дату (тип: `datetime.datetime | datetime.date`, по умолчанию: None) и возвращающую кверисет последних записей на указанную дату (на максимальную дату в таблице, если не указана) по переданному полю даты при группировке по переданным полям для группировки (то есть наиболее актуальную запись для каждой группы на указанную дату в виде кверисета).

Модель должна иметь публичную функцию, принимающую на вход дату (тип: `datetime.datetime | datetime.date`, по умолчанию: None) и возвращающую результат выполнения описанной выше функции кверисета. В качестве аргументов в функцию кверисета должны передаваться значение атрибута G (если отсутствует, возникает исключение), значение атрибута H (если отсутствует, то название поля W) и переданная дата.

4.2. Требования к временным характеристикам

Специальные требования не предъявляются.

4.3. Требования к интерфейсу

Специальные требования не предъявляются.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

4.4. Требования к надежности

Программа может выбрасывать любые исключения, которые выбрасывают функции фреймворка Django [14], вызываемые внутри ее функций. Также программа может выбрасывать конфигурационные исключения в случаях, упомянутых в подпунктах пункта 4.1, а также при нарушении типов ожидаемых значений полей и аргументов.

4.5. Условия эксплуатации

Требуется установленный на машине разработки язык Python [21], а также фреймворк Django [14].

4.6. Требования к составу и параметрам технических средств

Для работы с программой необходима любая машина, подходящая для разработки ПО, на которую можно установить язык Python [21], а также фреймворк Django [14].

4.7. Требования к информационной и программной совместимости

Программа должна быть написана на языке Python [21] версии 3.10 с использованием фреймворка Django [14] версии 4.1.6.

4.8. Требования к маркировке и упаковке

Специальные требования не предъявляются.

4.9. Требования к транспортировке и хранению

Специальные требования не предъявляются.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

5. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

5.1. Состав программной документации

- «Модуль утилитарных абстракций моделей для фреймворка Django». Техническое задание (ГОСТ 19.201-78 [7]);
- «Модуль утилитарных абстракций моделей для фреймворка Django». Пояснительная записка (ГОСТ 19.404-79 [10]);
- «Модуль утилитарных абстракций моделей для фреймворка Django». Программа и методика испытаний (ГОСТ 19.301-79 [11]);
- «Модуль утилитарных абстракций моделей для фреймворка Django». Текст программы (ГОСТ 19.401-78 [12]);
- «Модуль утилитарных абстракций моделей для фреймворка Django». Руководство программиста (ГОСТ 19.504-79 [13]);

5.2. Специальные требования к программной документации

- Документы к программе должны быть выполнены в соответствии с ГОСТ 19.106-78 и ГОСТами к каждому виду документа (см. 5.1.);
- Титульные листы всех документов должны иметь подписи академического руководителя, технического руководителя и студента;
- Все материалы документации должны быть загружены в системы SmartLMS не позднее даты загрузки материалов.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

6. ТЕХНИКО-ЭКОНОМИЧЕСКИЕ ПОКАЗАТЕЛИ

6.1. Ориентировочная экономическая эффективность

В рамках данной работы расчёт экономической эффективности не предусмотрен.

6.2. Предполагаемая потребность

Данная программа будет востребована среди всех разработчиков ПО, реализуемого с использованием фреймворка Django [14].

6.3. Преимущества разработки по сравнению с аналогами

Существует ряд пользовательских решений, находящихся в свободном доступе в виде SDK. Однако, даже при наличии в этих пакетах схожих по задумке абстрактных моделей их реализация существенно отличается. Данная программа предлагает свое решение, которое для некоторых пользователей окажется наиболее удобным. Основные преимущества решения (уникальные возможности, отсутствующие у аналогов): возможность логирования процесса обновления обновляемых моделей непосредственно в таблицах БД, предотвращение множественного перезапуска процесса обновления при помощи механизма логирования, автоматический контроль количества хранимых записей в таблицах логов, создание исторических таблиц с возможностью получить снимок данных на любую дату (актуальное состояние каждой отслеживаемой записи на дату) при помощи вызова функции с указанием нужной даты.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

7. СТАДИИ И ЭТАПЫ РАЗРАБОТКИ

Стадии и этапы разработки были выявлены с учетом ГОСТ 19.102-77 [2]:

Стадии разработки	Этапы работ	Содержание работ
1. Техническое задание	Подготовительные работы	Постановка задачи. Сбор теоретических сведений. Определение функциональных требований. Планирование архитектуры программы.
	Разработка и утверждение технического задания	Определение требований к программе. Определение требований к техническим средствам. Определение стадий, этапов и сроков разработки программы и документации на неё. Написание технического задания. Отправка технического задания, внесение корректировок. Согласование и утверждение технического задания.
2. Рабочий проект	Разработка программы	Разработка, отладка и тестирование программы.
	Разработка программной документации	Разработка программных документов в соответствии с требованиями ГОСТ 19.101-77.
	Испытания программы	Разработка, согласование и утверждение порядка и методики испытаний. Проведение испытаний.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

		Корректировка программы и программной документации по результатам испытаний.
3. Внедрение	Подготовка и передача программы	Подготовка и передача программы и программной документации для оценки.

Программа и документация к ней разрабатываются к утвержденным срокам защиты курсовой работы.

Исполнитель – Черемин Никита Сергеевич.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

8. ПОРЯДОК КОНТРОЛЯ И ПРИЕМКИ

8.1. Виды испытаний

Производится проверка корректности работы программы при моделировании потенциальных сценариев использования абстрактных моделей в коде разрабатываемого с ее помощью ПО. Проверяется соответствие программы техническому заданию. Осуществляется функциональное тестирование программы согласно «Программе и методике испытаний».

8.2. Общие требования к приемке работы

Прием программного продукта производится в сроки, указанные в пункте 7. при полной работоспособности программы, при выполнении указанных в п. 4.1. функций и при наличии документации к программе, указанной в п. 5.1, соответствующей требованиям пункта 5.2.

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. ГОСТ 19.101-77 Виды программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
2. ГОСТ 19.102-77 Стадии разработки. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
3. ГОСТ 19.103-77 Обозначения программ и программных документов. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
4. ГОСТ 19.104-78 Основные надписи. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
5. ГОСТ 19.105-78 Общие требования к программным документам. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
6. ГОСТ 19.106-78 Требования к программным документам, выполненным печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
7. ГОСТ 19.201-78 Техническое задание. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
8. ГОСТ 19.603-78 Общие правила внесения изменений. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
9. ГОСТ 19.604-78 Правила внесения изменений в программные документы, выполненные печатным способом. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
10. ГОСТ 19.404-79 Пояснительная записка. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
11. ГОСТ 19.301-79 Программа и методика испытаний. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
12. ГОСТ 19.401-78 Текст программы. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
13. ГОСТ 19.504-79 Руководство программиста. Требования к содержанию и оформлению. // Единая система программной документации. – М.: ИПК Издательство стандартов, 2001.
14. Django // [Электронный ресурс] — Режим доступа: свободный, URL: <https://www.djangoproject.com/> (04.02.2022)
15. Wikipedia // [Электронный ресурс] — Режим доступа: свободный, URL: <https://ru.wikipedia.org/wiki/SDK> (04.02.2022)
16. Wikipedia // [Электронный ресурс] — Режим доступа: свободный, URL: https://ru.wikipedia.org/wiki/Модель_данных (04.02.2022)
17. Wikipedia // [Электронный ресурс] — Режим доступа: свободный, URL: https://ru.wikipedia.org/wiki/Программное_обеспечение (04.02.2022)
18. Wikipedia // [Электронный ресурс] — Режим доступа: свободный, URL: <https://ru.wikipedia.org/wiki/ORM> (04.02.2022)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

19. Wikipedia // [Электронный ресурс] — Режим доступа: свободный, URL:
https://ru.wikipedia.org/wiki/База_данных
 (04.02.2022)
20. Wikipedia // [Электронный ресурс] — Режим доступа: свободный, URL:
https://ru.wikipedia.org/wiki/Объектно-ориентированное_программирование
 (04.02.2022)
21. Python // [Электронный ресурс] — Режим доступа: свободный, URL:
<https://www.python.org/>
 (04.02.2022)

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.06.05-01 ТЗ 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

Лист регистрации изменений

[illegible]