

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

Факультет компьютерных наук  
Образовательная программа «Программная инженерия»

СОГЛАСОВАНО

УТВЕРЖДАЮ

Технический руководитель  
внештатный преподаватель департамента  
программной инженерии

Академический руководитель образовательной  
программы «Программная инженерия»  
профессор департамента программной  
инженерии, канд. техн. наук



Н. И. Веселко  
«10» апреля 2023 г.

В. В. Шилов  
«10» апреля 2023 г.

**МОДУЛЬ УТИЛИТАРНЫХ АБСТРАКЦИЙ МОДЕЛЕЙ ДЛЯ ФРЕЙМВОРКА  
DJANGO**

**Программа и методика испытаний**

**ЛИСТ УТВЕРЖДЕНИЯ**

**RU.17701729.04.03-01 51 01-1-ЛУ**

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

Исполнитель  
студент группы БПИ203



Н. С. Черемин  
«10» апреля 2023 г.

Москва 2023

УТВЕРЖДЕН  
RU.17701729.04.03-01 51 01-1-ЛУ

**МОДУЛЬ УТИЛИТАРНЫХ АБСТРАКЦИЙ МОДЕЛЕЙ ДЛЯ ФРЕЙМВОРКА  
DJANGO**

**Программа и методика испытаний**

RU.17701729.04.03-01 51 01-1

**Листов 18**

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

**Москва 2023**

## СОДЕРЖАНИЕ

ГЛОССАРИЙ.....	4
1. ОБЪЕКТ ИСПЫТАНИЙ.....	5
1.1. Наименование программы.....	5
1.2. Краткая характеристика и область применения программы.....	5
2. ЦЕЛЬ ИСПЫТАНИЙ.....	6
3. ТРЕБОВАНИЯ К ПРОГРАММЕ .....	7
3.1. Требования к функциональным характеристикам.....	7
3.1.1. Требования к модели логов (LogModel) .....	7
3.1.2. Требования к модели логов обновления (UpdateLogModel).....	7
3.1.3. Требования к обновляемой логируемой модели (UpdatableLoggableModel).....	7
3.1.4. Требования к исторической модели (HistoryModel) .....	9
3.2. Требования к временным характеристикам.....	9
3.3. Требования к интерфейсу .....	9
3.4. Требования к надежности.....	10
4. ТРЕБОВАНИЯ К ПРОГРАММНОЙ ДОКУМЕНТАЦИИ.....	11
4.1. Состав программной документации.....	11
4.2. Специальные требования к программной документации .....	11
5. СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ.....	12
5.1. Технические средства, используемые во время испытаний.....	12
5.2. Программные средства, используемые во время испытаний .....	12
5.3. Порядок проведения испытаний.....	12
6. МЕТОДЫ ИСПЫТАНИЙ.....	13
6.1. Испытание выполнения требований к программной документации.....	13
6.2. Испытание требований к функциональным характеристикам и требований к надежности.....	13
6.2.1. Общие положения.....	13
6.2.2. Испытание модели логов (LogModel).....	13
6.2.3. Испытание модели логов обновления (UpdateLogModel).....	14
6.2.4. Испытание обновляемой логируемой модели (UpdatableLoggableModel).....	14
6.2.5. Испытание исторической модели (HistoryModel).....	14
СПИСОК ИСПОЛЪЗУЕМОЙ ЛИТЕРАТУРЫ.....	16

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

**ЛИСТ РЕГИСТРАЦИИ ИЗМЕНЕНИЙ.....18**

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 51 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`.
- Юнит-тестирование — процесс в программировании, позволяющий проверить на корректность отдельные модули исходного кода программы [25].
- Командный интерпретатор — компьютерная программа, часть операционной системы, обеспечивающая базовые возможности управления компьютером посредством интерактивного ввода команд через интерфейс командной строки или последовательного исполнения пакетных командных файлов [22].

Изм.	Лист	№ докум.	Подп.	Дата
RU.17701729.04.03-01 51 01-1				
Инв. № подл.	Подп. и дата	Взам. инв. №	Инв. № дубл.	Подп. и дата

## 1. ОБЪЕКТ ИСПЫТАНИЙ

### 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. ЦЕЛЬ ИСПЫТАНИЙ**

Целью испытаний является проверка правильности выполнения программой всех функций, перечисленных в разделе «Требования к программе».

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. ТРЕБОВАНИЯ К ПРОГРАММЕ

#### 3.1. Требования к функциональным характеристикам

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

##### 3.1.1. 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.

##### 3.1.2. 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).

##### 3.1.3. 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

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



обновления (тип: `datetime.timedelta`, по умолчанию: `None`) (далее: D), для настройки коэффициента 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 пустует 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

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

(если предыдущее обновление есть), вызывает защищенную функцию обновления в try-excerpt блоке и при возникновении исключения записывает его в сообщения лог в поле Y, затем при любых обстоятельствах записывает текущее время в поле Z.

#### 3.1.4. 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.

### 3.2. Требования к временным характеристикам

Специальные требования не предъявляются.

### 3.3. 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

### 3.4. 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.

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 ПРОГРАММНОЙ ДОКУМЕНТАЦИИ

##### 4.1. 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]);

##### 4.2. 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

## 5. СРЕДСТВА И ПОРЯДОК ИСПЫТАНИЙ

### 5.1. Технические средства, используемые во время испытаний

Состав технических средств: любое аппаратное обеспечение, на которое можно установить python [21] и сторонние библиотеки для него, которое имеет командный интерпретатор и к которому подключены источник ввода и источник графического вывода.

### 5.2. Программные средства, используемые во время испытаний

Работа программы была протестирована с использованием следующих программных средств:

1. Операционная система MacOS Monterey 12.1 [23];
2. Python 3.10 [21];
3. Django 4.1.6 [14];
4. PyCharm 2021.2 [24].

### 5.3. Порядок проведения испытаний

Испытания должны проводиться в следующем порядке:

1. Проверка требований к программной документации;
2. Проверка требований к функциональным характеристикам и требований к надежности.

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. МЕТОДЫ ИСПЫТАНИЙ

### 6.1. Испытание выполнения требований к программной документации

Программная документация визуально проверяется на предмет соответствия требованиям к оформлению и содержанию, на соответствие ГОСТам, на наличие подписей академического руководителя, технического руководителя и студента.

### 6.2. Испытание требований к функциональным характеристикам и требований к надежности

#### 6.2.1. Общие положения

Все требования к функциональным характеристикам и надежности испытаны при помощи юнит-тестов (представлены тест-кейсами), написанных с использованием встроенных в Django [14] средств юнит-тестирования (модуль `django.tests`). Для испытаний были написаны тестовые реализации абстрактных моделей (для демонстрации вариантов их использования). В каждом тест-кейсе содержатся тесты, имитирующие взаимодействие с данными реализациями и проверяющие корректность работы абстракций.

#### 6.2.2. Испытание модели логов (LogModel)

Тест-кейсы, покрывающие функционал модели: `DeletableModelTestCase`, `AutoDeletableModelTestCase`.

`DeletableModelTestCase` содержит функции:

- `test_objects_to_delete` — проверяет корректность получения записей, которые необходимо удалить, при различных значениях параметра максимального количества хранимых записей;
- `test_objects_to_delete_after_time_created_reversion` — проверяет корректность получения записей, которые необходимо удалить, при перестановке дат создания записей (подмене истории);
- `test_try_delete_objects` — проверяет корректность удаления старых записей при принудительном вызове функции удаления и при различных значениях параметра максимального количества хранимых записей.

`AutoDeletableModelTestCase` содержит функции:

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

- `test_auto_delete_after_create` — проверяет корректность автоматического удаления старых записей при создании новых;
- `test_delete_after_bulk_create` — проверяет корректность отработки функции единовременного множественного создания записей в сочетании с параметром максимального количества хранимых записей.

### 6.2.3. Испытание модели логов обновления (UpdateLogModel)

Тест-кейсы, покрывающие функционал модели, совпадают с тест-кейсами, покрывающими функционал LogModel.

### 6.2.4. Испытание обновляемой логируемой модели (UpdatableLoggableModel)

Тест-кейсы, покрывающие функционал модели: UpdatableModelTestCase, UpdatableLoggableModelTestCase.

UpdatableModelTestCase содержит функции:

- `test_is_updating_and_is_can_start_without_time_limit` — проверяет корректность работы функций проверки наличия текущего обновления и проверки возможности начала нового обновления в многопоточном режиме без установки параметра ограничения времени обновления;
- `test_is_updating_and_is_can_start_with_time_limit` — проверяет корректность работы функций проверки наличия текущего обновления и проверки возможности начала нового обновления в многопоточном режиме с установкой параметра ограничения времени обновления;
- `test_update_result_without_time_limit` — проверяет корректность работы функции обновления и ее возвращаемых значений в многопоточном режиме без установки параметра ограничения времени обновления;
- `test_update_result_with_time_limit` — проверяет корректность работы функции обновления и ее возвращаемых значений в многопоточном режиме с установкой параметра ограничения времени обновления.

UpdatableLoggableModelTestCase содержит функции:

- `test_saving_logs_and_returning_last_update_time_and_and_average_update_time` — проверяет корректность сохранения логов в процессе обновления и работы функций вычисления последнего времени обновления и среднего времени обновления.

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

**6.2.5. Испытание исторической модели (HistoryModel)**

Тест-кейсы, покрывающие функционал модели: HistoryModelTestCase.

HistoryModelTestCase содержит функции:

- test\_objects\_for\_date — проверяет корректность предоставления актуального состояния каждой строки на указанную дату (среди исторических записей, представляющих вариации каждой из строк).

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](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](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](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](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)
22. Wikipedia // [Электронный ресурс] — Режим доступа: свободный, URL:  
[https://ru.wikipedia.org/wiki/Интерпретатор\\_командной\\_строки](https://ru.wikipedia.org/wiki/Интерпретатор_командной_строки)  
(04.02.2022)
23. Apple // [Электронный ресурс] — Режим доступа: свободный, URL:  
<https://www.apple.com/ru/macOS/monterey/>  
(04.02.2022)
24. JetBrains // [Электронный ресурс] — Режим доступа: свободный, URL:  
<https://www.jetbrains.com/ru-ru/pycharm/>  
(04.02.2022)
25. Wikipedia // [Электронный ресурс] — Режим доступа: свободный, URL:  
[https://ru.wikipedia.org/wiki/Модульное\\_тестирование](https://ru.wikipedia.org/wiki/Модульное_тестирование)  
(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]