

**Федеральное государственное автономное образовательное  
учреждение высшего образования  
«Московский физико-технический институт  
(национальный исследовательский университет)»**

**УТВЕРЖДЕНО**

**Директор по цифровизации  
образования**

**Д.И. Гриц**

	<b>Рабочая программа дисциплины (модуля)</b>
<b>по дисциплине:</b>	Современные средства разработки
<b>по направлению:</b>	Бизнес-информатика
<b>профиль подготовки:</b>	Финансовые технологии и аналитика центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск" центр дополнительного, дополнительного профессионального и онлайн-образования "Пуск"
<b>курс:</b>	1
<b>квалификация:</b>	магистр

Семестры, формы промежуточной аттестации:

1 (осенний) - Зачет  
2 (весенний) - Зачет

Аудиторных часов: 135 всего, в том числе:

лекции: 60 час.  
семинары: 75 час.  
лабораторные занятия: 0 час.

Самостоятельная работа: 135 час.

Всего часов: 270, всего зач. ед.: 6

Программу составил: Р.С. Кулиев, старший преподаватель

Программа обсуждена на заседании центра дополнительного, дополнительного профессионального и онлайн-образования "Пуск" 30.06.2022

## Аннотация

Дисциплина состоит из двух модулей:

1 модуль. Основы программирования на Python.

2 модуль. Разработка на платформе 1С:Предприятие.

По итогам обучения обучающийся будет способен формализовать и алгоритмизировать поставленную задачу, написать программный код с использованием языков программирования, оформить код в соответствии с установленными требованиями.

## 1. Цели и задачи

### Цель дисциплины

Целью реализации дисциплины «Современные языки программирования»: формирование/совершенствование компетенций студентов в области решения профессиональных задач по программированию с использованием языков Python и 1С, применения основ программирования, в том числе асинхронного, на Python, проектирования программного обеспечения с помощью встроенного языка 1С.

### Задачи дисциплины

- сформировать умение использовать базовые типы и конструкции языка программирования Python;
- сформировать умение работать со стандартными структурами данных в Python, писать функции на Python, применять функциональные особенности языка, работать с файлами с помощью языка Python;
- сформировать умение применять механизмы наследования, создавать классы и работать с ними, обрабатывать исключения;
- сформировать умение искать и исправлять ошибки в программе на Python, тестировать программы на Python;
- сформировать умение писать многопоточный код на Python, писать асинхронный код на Python, работать с сетью, создать своё серверное сетевое приложение;
- изучить основные принципы, подходы и механизмы разработки бизнес-приложений на платформе 1С:Предприятие;
- изучить возможности быстрой кросс-платформенной разработки на платформе 1С:Предприятие;
- изучить возможности создания веб и мобильных приложений на платформе 1С:Предприятие.

## 2. Перечень формируемых компетенций

Освоение дисциплины направлено на формирование следующих компетенций:

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-1 Способен разрабатывать стратегию развития информационных технологий, инфраструктуры предприятия и управлять её реализацией	ОПК-1.1 Применяет на практике методики оценки качества ресурсов информационных технологий, управления активами и конфигурации информационных технологий, методики определения потребностей в уровне качества ресурсов ИТ

## 3. Перечень планируемых результатов обучения по дисциплине (модулю)

В результате освоения дисциплины обучающиеся должны знать:

- базовые сведения о языке, особенности организации кода на Python;
- стандартные структуры данных в Python;
- механизмы наследования, классы;
- особенности объектно-ориентированной модели в Python;
- процессы и потоки ОС;
- основные принципы, подходы и механизмы разработки бизнес-приложений на платформе 1С:Предприятие;
- возможности быстрой кросс-платформенной разработки на платформе 1С:Предприятие;
- возможности создания веб и мобильных приложений на платформе 1С:Предприятие.

уметь:

- использовать базовые типы и конструкции языка;
- работать со стандартными структурами данных в Python, писать функции на Python, применять функциональные особенности языка, работать с файлами с помощью языка Python;
- применять механизмы наследования, создавать классы и работать с ними, обрабатывать исключения;
- искать и исправлять ошибки в программе на Python, тестировать программы на Python;
- писать многопоточный код на Python, писать асинхронный код на Python, работать с сетью, создать своё серверное сетевое приложение;
- работать в команде;
- создавать кросс-платформенные приложения на платформе 1С:Предприятие;
- разрабатывать веб и мобильные приложения на платформе 1С:Предприятие;
- самостоятельно применять языки программирования (в т.ч. скрипты) и настраиваемые программные инструменты для автоматизации процессов в профессиональной деятельности.

владеть:

- навыками разработки алгоритмов и программ, пригодных для практического применения;
- навыками освоения методик использования программных средств для решения практических задач;
- навыками написания программного кода с использованием языков программирования, оформления кода в соответствии с установленными требованиями.

#### 4. Содержание дисциплины (модуля), структурированное по темам (разделам) с указанием отведенного на них количества академических часов и видов учебных занятий

##### 4.1. Разделы дисциплины (модуля) и трудоемкости по видам учебных занятий

№	Тема (раздел) дисциплины	Трудоемкость по видам учебных занятий, включая самостоятельную работу, час.			
		Лекции	Семинары	Лаборат. работы	Самост. работа
1	Основы программирования на Python	30	30		75
2	Разработка на платформе 1С:Предприятие	30	45		60
Итого часов		60	75		135
Подготовка к экзамену		0 час.			
Общая трудоёмкость		270 час., 6 зач.ед.			

##### 4.2. Содержание дисциплины (модуля), структурированное по темам (разделам)

Семестр: 1 (Осенний)

#### 1. Основы программирования на Python

Основы программирования на Python. Структуры данных и функции. Объектно-ориентированное программирование. Углубленный Python. Многопоточное и асинхронное программирование.

Семестр: 2 (Весенний)

## 2. Разработка на платформе 1С:Предприятие

Платформенный подход к разработке бизнес-приложений. Основные типы и элементы платформы. Основные механизмы регистрации и хранения показателей. Управляемые формы. Вывод данных. Механизмы интеграции. Механизмы коллаборации и коммуникации. Мобильная платформа. Веб-клиент. Обслуживание и эксплуатация информационной системы.

## 5. Описание материально-технической базы, необходимой для осуществления образовательного процесса по дисциплине (модулю)

Система дистанционного обучения:

Обучающемуся необходимо наличие доступа в сеть интернет, компьютер.

Преподавателю курса необходимо наличие доступа администратора курса и оборудование для проведения дистанционных семинаров (вебинаров), качественный отказоустойчивый доступ в сеть интернет.

## 6.Перечень рекомендуемой литературы

Основная литература

1. Программирование на языке высокого уровня Python , учебное пособие для вузов/Д. Ю. Федоров,-Москва, Юрайт, 2020

Дополнительная литература

1. Python и анализ данных, Электрон. версия печ. публикации / У. Маккини. — Москва, ДМК Пресс, 2020
2. Python и анализ данных, Первичная обработка данных с применением pandas, NumPy и IPython / У. Маккини. — Москва, ДМК Пресс, 2020.— URL: <https://e.lanbook.com/book/131721> (дата обращения: 26.01.2021). - Полный текст (Режим доступа : из сети МФТИ / Удаленный доступ)
3. Статистика для всех / С. Бослаф. — Москва, ДМК Пресс, 2015.— URL: <https://e.lanbook.com/book/66475> (дата обращения: 26.01.2021). - Полный текст (Режим доступа : из сети МФТИ / Удаленный доступ)
4. Теория вероятностей и математическая статистика в задачах [Текст] : учеб. пособие для вузов / В. А. Ватутин [и др.] .— 3-е изд., испр. — М. : Дрофа, 2005 .— 315 с.

## 7. Перечень ресурсов информационно-телекоммуникационной сети "Интернет", необходимых для освоения дисциплины (модуля)

Think Python [Электронный ресурс] – Режим доступа -

<https://greenteapress.com/wp/think-python-2e/>

Automate the Boring Stuff with Python [Электронный ресурс] – Режим доступа -

<https://automatetheboringstuff.com/>

Dive Into Python 3 [Электронный ресурс] – Режим доступа

-<http://diveintopython3.problemsolving.io/>

Problem Solving with Algorithms and Data Structures using Python [Электронный ресурс] – Режим доступа -<https://runestone.academy/runestone/static/pythonds/index.html>

Swaroop Chitlur. A Byte of Python [Электронный ресурс] – Режим доступа - <https://wombat.org.ua/AByteOfPython/AByteofPythonRussian-2.02.pdf> – 2020.

Справочник по функциям DAX [Электронный ресурс] – Режим доступа -

<https://docs.microsoft.com/ru-ru/dax/dax-function-reference>

**8. Перечень информационных технологий, используемых при осуществлении образовательного процесса по дисциплине (модулю), включая перечень необходимого программного обеспечения и информационных справочных систем (при необходимости)**

Документация Postgres про сравнение строк -

<https://postgrespro.ru/docs/postgrespro/9.5/functions-matching>

Документация Postgres про другие функции работы со строками -

<https://postgrespro.ru/docs/postgrespro/9.5/functions-string>

Тестер регулярных выражений - <https://www.regextester.com>

Интерактивный учебник по SQL - <http://www.sql-tutorial.ru/ru/content.html>

Введение в анализ данных с помощью Pandas - <https://habr.com/ru/post/196980/>

Начало работы с Power BI -

<https://docs.microsoft.com/ru-ru/power-bi/fundamentals/desktop-getting-started>

Интернет ресурс: <http://v8.1c.ru>

**9. Методические указания для обучающихся по освоению дисциплины (модуля)**

Самостоятельная работа подразделяется на аудиторную и внеаудиторную. Аудиторную самостоятельную работу составляют практические задания, которые выполняются слушателями во время учебных занятий, результаты ее выполнения проверяются и оцениваются преподавателем в учебном процессе.

Внеаудиторная самостоятельная работа включает формы: изучение дополнительной литературы, подготовка итоговых проектов по модулям, подготовка проекта.

Основными критериями качества организации самостоятельной работы служит наличие контроля результатов самостоятельной работы.

Основными современными формами организации самостоятельной работы являются творческие работы и работа с информационными компьютерными технологиями.

**ОЦЕНОЧНЫЕ МАТЕРИАЛЫ ПО ДИСЦИПЛИНЕ (МОДУЛЮ)**

<b>по направлению:</b>	Бизнес-информатика		
<b>профиль подготовки:</b>	Финансовые технологии и аналитика	▲	▲
	онлайн-образования "Пуск"	▲	▲
	онлайн-образования "Пуск"		
<b>курс:</b>	1		
<b>квалификация:</b>	магистр		
Семестры, формы промежуточной аттестации:			
	1 (осенний) - Зачет		
	2 (весенний) - Зачет		
<b>Разработчик:</b>	Р.С. Кулиев, старший преподаватель		

## 1. Компетенции, формируемые в процессе изучения дисциплины

Код и наименование компетенции	Индикаторы достижения компетенции
ОПК-1 Способен разрабатывать стратегию развития информационных технологий, инфраструктуры предприятия и управлять её реализацией	ОПК-1.1 Применяет на практике методики оценки качества ресурсов информационных технологий, управления активами и конфигурации информационных технологий, методики определения потребностей в уровне качества ресурсов ИТ

## 2. Показатели оценивания компетенций

В результате изучения дисциплины «Современные средства разработки» обучающийся должен:

### знать:

- базовые сведения о языке, особенности организации кода на Python;
- стандартные структуры данных в Python;
- механизмы наследования, классы;
- особенности объектно-ориентированной модели в Python;
- процессы и потоки ОС;
- основные принципы, подходы и механизмы разработки бизнес-приложений на платформе 1С:Предприятие;
- возможности быстрой кросс-платформенной разработки на платформе 1С:Предприятие;
- возможности создания веб и мобильных приложений на платформе 1С:Предприятие.

### уметь:

- использовать базовые типы и конструкции языка;
- работать со стандартными структурами данных в Python, писать функции на Python, применять функциональные особенности языка, работать с файлами с помощью языка Python;
- применять механизмы наследования, создавать классы и работать с ними, обрабатывать исключения;
- искать и исправлять ошибки в программе на Python, тестировать программы на Python;
- писать многопоточный код на Python, писать асинхронный код на Python, работать с сетью, создать своё серверное сетевое приложение;
- работать в команде;
- создавать кросс-платформенные приложения на платформе 1С:Предприятие;
- разрабатывать веб и мобильные приложения на платформе 1С:Предприятие;
- самостоятельно применяет языки программирования (в т.ч. скрипты) и настраиваемые программные инструменты для автоматизации процессов в профессиональной деятельности.

### владеть:

- навыками разработки алгоритмов и программ, пригодных для практического применения;
- навыками освоения методик использования программных средств для решения практических задач;
- навыками написания программного кода с использованием языков программирования, оформления кода в соответствии с установленными требованиями.

## 3. Перечень типовых (примерных) вопросов, заданий, тем для подготовки к текущему контролю

С целью контроля освоения обучающимися учебного материала проводится устный опрос в начале занятия по теме прошлого занятия.

## 4. Перечень типовых (примерных) вопросов и тем для проведения промежуточной аттестации обучающихся

Разработать бизнес-приложение в соответствии с следующим сценарием:

Установить систему, создать конфигурацию, сделать простейшую записную книжку.

Расширить записную книжку, добавив возможность создания списка комментариев к контакту, а также загрузки фотографии контакта.

Создать простейшую систему автоматизации интернет-магазина.

Добавить возможность создания продаж контактам из записной книжки.

Добавить удобный интерфейс пользователя системы.

Добавить отчеты о продажах в разрезах товара, группы товаров, контактов, временных интервалов.

Добавить построение диаграмм продаж, в том числе, в разрезе по отдельным менеджерам.

Организовать выгрузку продаж в табличном формате с возможностью выбора столбцов в выгрузке.

Добавить возможность выставления цены в валюте (путем интеграции с ЦБ РФ для получения курсов валют).

Создать чат-бота для клиентов (для обсуждения заказов).

Добавить возможность вести контекстные обсуждения по заказам для пользователей системы.

Создать мобильное приложение для разработанной конфигурации.

Опубликовать свое приложение как веб-сервис. Добавить возможность установки как PWA.

Добавить пользователей системы. Реализовать возможность восстановления пароля пользователем.

#### Критерии оценивания

Оценка «зачтено» выставляется студенту, если он показал всесторонние, систематизированные, глубокие знания учебной программы дисциплины и умение уверенно применять их на практике при решении конкретных задач, свободное и правильное обоснование принятых решений.

Оценка «не зачтено» выставляется студенту, который не знает большей части основного содержания учебной программы дисциплины, допускает грубые ошибки в формулировках основных понятий дисциплины и не умеет использовать полученные знания при решении типовых практических задач.

#### **5. Методические материалы, определяющие процедуры оценивания знаний, умений, навыков и (или) опыта деятельности**

Во время проведения зачета обучающиеся могут пользоваться программой дисциплины.



## Примеры заданий по модулю 1

### Практические задания

#### Тема 1. Введение в Python

Написать программу (скрипт), которая будет запускаться из командной строки. Программа принимает в качестве аргумента строку, состоящую из цифр. Гарантируется, что других символов в переданном параметре нет и на вход всегда подается не пустая строка. Программа должна вычислить сумму цифр из которых состоит строка и вывести полученный результат на печать в стандартный вывод.

#### Тема 2. Структуры данных и функции

Реализовать собственное **key-values** хранилище. Данные будут сохраняться в файле **storage.data**. Добавление новых данных в хранилище и получение текущих значений осуществляется с помощью утилиты командной строки **storage.py**.

#### Тема 3. Объектно-ориентированное программирование

Написать python-модуль **solution.py**, внутри которого необходимо поместить код класса **FileReader**. Конструктор этого класса принимает один параметр: путь до файла на диске. В классе **FileReader** должен быть реализован метод **read**, возвращающий строку - содержимое файла, путь к которому был указан при создании экземпляра класса. Python модуль должен быть написан таким образом, чтобы импорт класса **FileReader** из него не вызвал ошибок.

При написании реализации метода **read**, вам нужно учитывать случай, когда при инициализации был передан путь к несуществующему файлу. Требуется обработать возникающее при этом исключение **FileNotFoundError** и вернуть из метода **read** пустую строку.

#### Тема 4. Углубленный Python

Создать интерфейс для работы с файлами. Интерфейс должен предоставлять следующие возможности по работе с файлами:

- чтение из файла, метод **read** возвращает строку с текущим содержанием файла
- запись в файл, метод **write** принимает в качестве аргумента строку с новым содержанием файла
- сложение объектов типа **File**, результатом сложения является объект класса **File**, при этом создается новый файл и файловый объект, в котором содержимое второго файла добавляется к содержимому первого файла. Новый файл должен создаваться в директории, полученной с помощью функции [tempfile.gettempdir](#). Для получения нового пути можно использовать [os.path.join](#).
- возвращать в качестве строкового представления объекта класса **File** полный путь до файла
- поддерживать протокол итерации, причем итерация проходит по строкам файла

При создании экземпляра класса **File** в конструктор передается полный путь до файла на файловой системе. Если файла с таким путем не существует, он должен быть создан при инициализации.

#### Тема 5. Многопоточное и асинхронное программирование

В крупных проектах, с большим количеством пользователей, необходимо тщательно наблюдать за всеми процессами, происходящими в нем. Информация о процессах может быть представлена различными численными показателями, например: количество запросов к вашему приложению, время ответа вашего сервиса на каждый запрос, количество

пользователей в сутки и другие. Эти различные численные показатели мы будем называть метриками.

Для сбора, хранения и отображения подобных метрик существуют готовые решения, например Graphite, InfluxDB. Мы в рамках курса разработаем свою систему для сбора и хранения метрик, основанную на клиент-серверной архитектуре.

На этой неделе мы начнем с разработки клиента для отправки и получения метрик. На следующей неделе, в качестве финального задания, вам будет предложено реализовать сервер для хранения метрик.

### ***Протокол взаимодействия***

Прежде, чем приступить к описанию задания, рассмотрим протокол взаимодействия, по которому будет происходить обмен данными между клиентом и сервером.

Клиент и сервер взаимодействуют между собой по простому текстовому протоколу через TCP сокет. Текстовый протокол имеет главное преимущество – наглядность, можно просматривать диалог взаимодействия клиентской и серверной стороны без использования дополнительных инструментов.

Общий формат запросов и ответов.

Протокол поддерживает два вида запросов к серверу со стороны клиента:

- отправка данных для сохранения их на сервере
- получения сохраненных данных

Общий формат запроса клиента:

```
<команда> <данные запроса><\n>
```

где:

- <команда> - команда сервера (команда может принимать одно из двух значений: put — сохранить данные на сервере, get — вернуть сохраненные данные с сервера),

- <данные запроса> - данные запроса (их формат мы подробно разберем ниже в примере),

- <\n> - символ переноса строки.

Обратим ваше внимание на пробел между командой и данными запроса и его отсутствием между данными и символом перевода на новую строку.

Общий формат ответов сервера:

```
<статус ответа><\n><данные ответа><\n\n>
```

где:

- <статус ответа> - статус выполнения команды, допустимы два варианта: «ok» - команда успешно выполнена на сервере и «error» - выполнение команды завершилось ошибкой

- <данные ответа> - не обязательное поле (формат ответа и случаи его отсутствия будут рассмотрены в примере ниже)

- <\n\n> - два символа переноса строки.

Обратите внимание, что статус ответа и данные ответа разделены символом перевода строки <\n>.

### ***Метод put***

Метод **put** принимает в качестве параметров: название метрики, численное значение и необязательный именованный параметр **timestamp**. Если пользователь вызвал метод **put** без аргумента **timestamp**, то клиент автоматически должен подставить значение временной отметки, полученное с помощью вызова **int(time.time())**.

Метод **put** не возвращает ничего в случае успешной отправки и выбрасывает пользовательское исключение **ClientError** в случае не успешной.

### ***Метод get***

Метод **get** принимает в качестве параметра имя метрики, значения которой мы хотим получить. В качестве имени метрики можно использовать символ «\*», о котором мы упоминали в описании протокола.

Метод **get** возвращает словарь с метриками (смотрите пример ниже) в случае успешного получения ответа от сервера и выбрасывает исключение **ClientError** в случае не успешного.

Клиент получает данные от сервера в текстовом виде, метод **get** должен обработать строку ответа и вернуть словарь с полученными ключами с сервера. Значением ключей в словаре является список кортежей:

```
[(timestamp1, metric_value1), (timestamp2, metric_value2), ...]
```

Значение `timestamp` и `metric_value` должны быть преобразованы соответственно к типам `int` и `float`. Список должен быть отсортирован по значению `timestamp` (по возрастанию).

## **Примеры заданий по модулю 2**

### ***Практические задания***

Тема 1. Установка системы, варианты поставки программы. Пример быстрой разработки приложения «с нуля до отчетов».

Тема 2. Работа с регистром сведений. Работа с конструктором управляемых форм. Конструирование интерфейса на подсистемах.

Тема 3. Работа с документами.

Тема 4. Работа с формами. Создание форм без использования модальности.

Тема 5. Работа с запросами и извлечение данных из базы, работа со сложными запросами, соединения и объединения запросов, формирование и настройка отчетов, формирование отчетов с диаграммами.

Тема 6. Формирование планов обмена, организация выгрузки данных в табличные файлы, настройка интеграции с веб-сервисами, работа с внешними компонентами.

Тема 7. Работа с элементами системы взаимодействия, создание чат-ботов, контекстных обсуждений.

Тема 8. Пример создания простейшего мобильного приложения.

Тема 9. Настройка веб-сервера. Публикация конфигурации на веб-сервере. Применение технологии PWA.

Тема 10. Выполнение регламентных операций с информационной базой. Настройка пользователей системы и прав доступа.

### ***Итоговый проект***

Разработать бизнес-приложение в соответствии с следующим сценарием:

1. Установить систему, создать конфигурацию, сделать простейшую записную книжку.
2. Расширить записную книжку, добавив возможность создания списка комментариев к контакту, а также загрузки фотографии контакта.
3. Создать простейшую систему автоматизации интернет-магазина.
4. Добавить возможность создания продаж контактам из записной книжки.

5. Добавить удобный интерфейс пользователя системы.
6. Добавить отчеты о продажах в разрезах товара, группы товаров, контактов, временных интервалов.
7. Добавить построение диаграмм продаж, в том числе, в разрезе по отдельным менеджерам.
8. Организовать выгрузку продаж в табличном формате с возможностью выбора столбцов в выгрузке.
9. Добавить возможность выставления цены в валюте (путем интеграции с ЦБ РФ для получения курсов валют).
10. Создать чат-бота для клиентов (для обсуждения заказов).
11. Добавить возможность вести контекстные обсуждения по заказам для пользователей системы.
12. Создать мобильное приложение для разработанной конфигурации.
13. Опубликовать свое приложение как веб-сервис. Добавить возможность установки как PWA.
14. Добавить пользователей системы. Реализовать возможность восстановления пароля пользователем.

В зачетно-экзаменационную ведомость оценка выставляется в соответствии с нижеприведенной таблицей 1.

Таблица 1

Сумма баллов	Оценка
50-100	Зачтено
менее 50	Не зачтено

Составляющие процесса обучения, которые оцениваются в ходе обучения, и их вклад в итоговую оценку представлены в таблице 2.

Таблица 2

№ п/п	Основные показатели оценки	Вклад в итоговую оценку
1	Практические занятия	30%
2	Выполнение домашних заданий	40%
3	Промежуточная аттестация	30%

### 3.2. Оценочные материалы

Таблица 3

Наименование модуля	Формы и методы контроля и оценки	Вес задания
Основы программирования на Python	Практические задания по темам лекций Домашние задания в системе дистанционного обучения по темам лекций	14
Структуры данных и функции	Практические задания по темам лекций Домашние задания в системе дистанционного обучения по темам лекций	14
Объектно-ориентированное программирование	Практические задания по темам лекций Домашние задания в системе дистанционного обучения по темам лекций	14
Углубленный Python	Практические задания по темам лекций Домашние задания в системе дистанционного обучения по темам лекций	14
Многопоточное и асинхронное программирование	Практические задания по темам лекций Домашние задания в системе дистанционного обучения по темам лекций	14
Промежуточная аттестация	Итоговое тестирование	30