

МИНОБРНАУКИ РОССИИ



Федеральное государственное автономное образовательное учреждение
высшего образования
«**Российский государственный гуманитарный университет**»
(ФГАОУ ВО «РГГУ»)

ИНСТИТУТ ЛИНГВИСТИКИ
Учебно-научный центр компьютерной лингвистики

Объектно-ориентированное программирование в Python

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ

45.04.03 Фундаментальная и прикладная лингвистика

Код и наименование направления подготовки/специальности

Фундаментальная и компьютерная лингвистика

Наименование направленности (профиля)/ специализации

Уровень высшего образования: магистратура

Форма обучения: очная

РПД адаптирована для лиц
с ограниченными возможностями
здоровья и инвалидов

Москва 2024

Объектно-ориентированное программирование в Python
Рабочая программа дисциплины

Составитель(и):

старший преподаватель А.М.Ивойлова

Ответственный редактор:

к.ф.н, доцент Н.А.Коротаев

УТВЕРЖДЕНО

Протокол заседания УНЦ компьютерной лингвистики

№ 5 от 26 марта 2024 г.

ОГЛАВЛЕНИЕ

1.	Пояснительная записка	
1.1.	Цель и задачи дисциплины	
1.2.	Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций	
1.3.	Место дисциплины в структуре образовательной программы	
2.	Структура дисциплины	
3.	Содержание дисциплины	
4.	Образовательные технологии	
5.	Оценка планируемых результатов обучения	
5.1	Система оценивания	
5.2	Критерии выставления оценки по дисциплине	
5.3	Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине	
6.	Учебно-методическое и информационное обеспечение дисциплины	
6.1	Список источников и литературы	
6.2	Перечень ресурсов информационно-телекоммуникационной сети «Интернет».	
6.3	Профессиональные базы данных и информационно-справочные системы	
7.	Материально-техническое обеспечение дисциплины	
8.	Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья и инвалидов	
9.	Методические материалы	
9.1	Планы семинарских/ практических/ лабораторных занятий	
9.2	Методические рекомендации по подготовке письменных работ	
9.3	Иные материалы	

1. Пояснительная записка

1.1. Цель и задачи дисциплины

Предметом дисциплины является углубленное изучение парадигмы объектно-ориентированного программирования и инструментов ООП в языке Python. В результате изучения курса студенты получают полные знания о разработке приложений в парадигме ООП, научатся использовать основные методы и алгоритмы разработки, познакомятся с такими понятиями, как наследование, инкапсуляция, полиморфизм, научатся самостоятельно создавать дескрипторы атрибутов, декораторы функций и классов, перегружать методы классов. Знания, полученные в результате освоения данного курса, будут очень полезны как при разработке собственных приложений и библиотек для автоматической обработки естественного языка, так и при использовании готовых инструментов.

Курс направлен на решение следующих задач:

- познакомить магистрантов с парадигмой объектно-ориентированного программирования и ее реализацией на языке Python;
- научить магистрантов выбирать и правильно использовать подходящие инструменты при разработке приложений и библиотек;
- научить магистрантов анализировать результаты применения алгоритмов программирования к лингвистическим данным;
- дать магистрантам знания, позволяющие им квалифицированно читать литературу по специальности, включающую в себя как учебные материалы и научные статьи, так и более специализированные технические материалы, например, программную документацию.

1.2. Перечень планируемых результатов обучения по дисциплине, соотнесенных с индикаторами достижения компетенций

Компетенция	Индикаторы компетенций	Результаты обучения
ПК-3 Способен использовать лингвистические технологии для проектирования систем автоматической обработки звучащей речи и письменного текста на естественном языке, лингвистических компонентов интеллектуальных и информационных электронных систем	ПК-3.2 Имеет практический опыт работы с системами автоматической обработки текста и звучащей речи; проектирования модулей таких систем	Знать: <ul style="list-style-type: none"> – принципы работы лингвистически ориентированных программных продуктов; Уметь: <ul style="list-style-type: none"> – пользоваться лингвистически ориентированными программными продуктами; Владеть: <ul style="list-style-type: none"> – навыками использования лингвистически ориентированных программных продуктов.

1.3. Место дисциплины в структуре образовательной программы

Дисциплина «Объектно-ориентированное программирование в Python» является элективной дисциплиной и относится к части, формируемой участниками образовательных отношений.

Для освоения дисциплины необходимы знания, умения и владения, сформированные в ходе изучения следующих дисциплин и прохождения практик: Основы языка программирования Python, Инструменты лингвистического анализа в Python.

2. Структура дисциплины

Общая трудоёмкость дисциплины составляет 2 з.е., 72 академических часа.

Структура дисциплины для очной формы обучения

Объем дисциплины в форме контактной работы обучающихся с педагогическими работниками и (или) лицами, привлекаемыми к реализации образовательной программы на иных условиях, при проведении учебных занятий:

Семестр	Тип учебных занятий	Количество часов
3	Практические занятия	20
Всего:		20

Объем дисциплины (модуля) в форме самостоятельной работы обучающихся составляет 52 академических часа.

3. Содержание дисциплины

№ п/п	Наименование раздела дисциплины	Содержание
1.	Принципы объектно-ориентированного программирования	<ul style="list-style-type: none"> • Основные понятия объектно-ориентированного программирования: наследование, полиморфизм, инкапсуляция. Основные принципы ООП: SOLID. • Определение пользовательских классов в языке Python. Атрибуты и методы классов. Статические и динамические атрибуты. Методы экземпляра класса. • Написание программ с использованием пользовательских классов и наследованием.
2.	Перегрузка методов	<ul style="list-style-type: none"> • Методы экземпляра класса и магические методы экземпляра класса. Понятие перегрузки методов. Протокол итерации. Магические методы, связанные с протоколом итерации. • Переопределение магических методов для итерирования.
3.	Наследование	<ul style="list-style-type: none"> • Наследование в Python. Понятие родительского класса. Множественное наследование. Порядок разрешения методов: MRO. • Альтернатива наследованию: композиция

		<p>классов.</p> <ul style="list-style-type: none"> • Написание программ с использованием множественного наследования и подмешивания классов.
4.	Инкапсуляция	<ul style="list-style-type: none"> • Понятие инкапсуляции. Интерфейс класса. Особенности реализации инкапсуляции в языке Python. • Псевдозакрытые атрибуты и методы. • Написание программ с использованием классов с инкапсуляцией.
5.	Дескрипторы. Исключения	<ul style="list-style-type: none"> • Понятие дескриптора. Имплементация дескрипторов в языке Python. Реализация дескрипторов с помощью функций и классов. Property. • Исключения: обработка исключений. Оператор try\except. Создание собственных исключений. • Написание программ с перехватом исключений и использованием дескрипторов атрибутов класса.
6.	Декораторы	<ul style="list-style-type: none"> • Понятие декоратора. Синтаксис декоратора в Python. Closure. • Методы класса и статические методы. • Написание программ с использованием декораторов для создания статических методов, методов класса и дескрипторов. Написание фабрики декораторов.
7.	Метаклассы	<ul style="list-style-type: none"> • Понятие метакласса. Создание пользовательских классов с использованием функции type(). Абстрактные классы и методы. • Написание программ с использованием метаклассов и абстрактных классов.

4. Образовательные технологии

Для проведения учебных занятий по дисциплине используются различные образовательные технологии. Для организации учебного процесса может быть использовано электронное обучение и (или) дистанционные образовательные технологии.

5. Оценка планируемых результатов обучения

5.1 Система оценивания

Форма контроля	Макс. количество баллов	
	За одну работу	Всего
Текущий контроль:		
- домашние задания	5 баллов	30 баллов
- выполнение заданий на семинаре	5 баллов	10 баллов
- проект	20 баллов	20 баллов

Промежуточная аттестация – зачет с оценкой	40 баллов
Итого за семестр	100 баллов

Полученный совокупный результат конвертируется в традиционную шкалу оценок и в шкалу оценок Европейской системы переноса и накопления кредитов (European Credit Transfer System; далее – ECTS) в соответствии с таблицей:

100-балльная шкала	Традиционная шкала		Шкала ECTS
95 – 100	отлично	зачтено	A
83 – 94			B
68 – 82	хорошо		C
56 – 67	удовлетворительно		D
50 – 55			E
20 – 49	неудовлетворительно	не зачтено	FX
0 – 19			F

5.2 Критерии выставления оценки по дисциплине

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
100-83/ A,B	отлично/ зачтено	<p>Выставляется обучающемуся, если он глубоко и прочно усвоил теоретический и практический материал, может продемонстрировать это на занятиях и в ходе промежуточной аттестации.</p> <p>Обучающийся исчерпывающе и логически стройно излагает учебный материал, умеет увязывать теорию с практикой, справляется с решением задач профессиональной направленности высокого уровня сложности, правильно обосновывает принятые решения.</p> <p>Свободно ориентируется в учебной и профессиональной литературе.</p> <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «высокий».</p>
82-68/ C	хорошо/ зачтено	<p>Выставляется обучающемуся, если он знает теоретический и практический материал, грамотно и по существу излагает его на занятиях и в ходе промежуточной аттестации, не допуская существенных неточностей.</p> <p>Обучающийся правильно применяет теоретические положения при решении практических задач профессиональной направленности разного уровня сложности, владеет необходимыми для этого навыками и приёмами.</p> <p>Достаточно хорошо ориентируется в учебной и профессиональной литературе.</p> <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «хороший».</p>
67-50/ D,E	удовлетворительно/ зачтено	<p>Выставляется обучающемуся, если он знает на базовом уровне теоретический и практический материал, допускает отдельные ошибки при его изложении на занятиях и в ходе промежуточной аттестации.</p> <p>Обучающийся испытывает определённые затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, владеет необходимыми для этого базовыми навыками и приёмами.</p> <p>Демонстрирует достаточный уровень знания учебной литературы по дисциплине.</p> <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции, закреплённые за дисциплиной, сформированы на уровне – «достаточный».</p>

Баллы/ Шкала ECTS	Оценка по дисциплине	Критерии оценки результатов обучения по дисциплине
49-0/ F,FX	неудовлет- ворительно/ не зачтено	<p>Выставляется обучающемуся, если он не знает на базовом уровне теоретический и практический материал, допускает грубые ошибки при его изложении на занятиях и в ходе промежуточной аттестации.</p> <p>Обучающийся испытывает серьёзные затруднения в применении теоретических положений при решении практических задач профессиональной направленности стандартного уровня сложности, не владеет необходимыми для этого навыками и приёмами.</p> <p>Демонстрирует фрагментарные знания учебной литературы по дисциплине.</p> <p>Оценка по дисциплине выставляется обучающемуся с учётом результатов текущей и промежуточной аттестации.</p> <p>Компетенции на уровне «достаточный», закреплённые за дисциплиной, не сформированы.</p>

5.3 Оценочные средства (материалы) для текущего контроля успеваемости, промежуточной аттестации обучающихся по дисциплине

В качестве домашних заданий предлагаются задания следующих типов

- Д31. Разработка собственной игры в Python с использованием ООП
- Д32. Создание голосового помощника в Python
- Д33. Написание графических интерфейсов к консольным инструментам обработки естественного языка
- Д34. Создание программы для хранения базы данных
- Д35. Проект на платформе Github: командная разработка любого инструмента с использованием ООП

Экзамен ориентирован на следующие контрольные вопросы

Основные понятия объектно-ориентированного программирования. Наследование, инкапсуляция, полиморфизм.

Атрибуты и методы. Виды атрибутов, виды методов. Магические методы.

Принципы SOLID.

Наследование: родительский класс, наследование методов, множественное наследование.

Инкапсуляция: интерфейс класса, псевдозакрытые атрибуты, property, особенности реализации методов инкапсуляции в Python.

Перегрузка методов. Протокол итерации. Проверка членства.

Дескрипторы. Синтаксис дескрипторов в Python.

Декораторы функций и классов. Замыкание (closure). Фабрики декораторов.

Метаклассы.

6. Учебно-методическое и информационное обеспечение дисциплины

6.1 Список источников и литературы

Основная литература

1. Марк Лутц. "Изучаем Python". Том 2. 5-е издание. 2019
2. Официальная документация (эталонная реализация CPython) языка программирования Python версии 3.x.x: <https://docs.python.org/3/>
3. Марк Саммерфилд, "Программирование на Python 3. Подробное руководство"

Рекомендованная литература

1. Rentsch, T. (1982). Object oriented programming. ACM Sigplan Notices, 17(9), 51-57.
2. Wegner, P. (1990). Concepts and paradigms of object-oriented programming. ACM Sigplan Oops Messenger, 1(1), 7-87.
3. Phillips, D. (2015). Python 3 object-oriented programming. Packt Publishing Ltd.
4. Miller, C. S., Settle, A., & Lalor, J. (2015, September). Learning object-oriented programming in python: Towards an inventory of difficulties and testing pitfalls. In Proceedings of the 16th annual conference on information technology education (pp. 59-64).

6.2 Перечень ресурсов информационно-телекоммуникационной сети «Интернет»

№ п/п	Наименование
1	Международные реферативные наукометрические БД, доступные в рамках национальной подписки в 2020 г. Web of Science Scopus
2	Профессиональные полнотекстовые БД, доступные в рамках национальной подписки в 2020 г. Журналы Cambridge University Press ProQuest Dissertation & Theses Global SAGE Journals Журналы Taylor and Francis
3	Профессиональные полнотекстовые БД JSTOR Издания по общественным и гуманитарным наукам Электронная библиотека Grebennikon.ru

6.3 Профессиональные базы данных и информационно-справочные системы

Доступ к профессиональным базам данных: <https://liber.rsuh.ru/ru/bases>

Официальная документация языка Python: <https://docs.python.org/3/>

7. Материально-техническое обеспечение дисциплины

№п /п	Наименование ПО	Производитель	Способ распространения (лицензионное или свободно распространяемое)
1	Microsoft Share Point 2010	Microsoft	лицензионное
2	Windows 10 Pro	Microsoft	лицензионное
3	Kaspersky Endpoint Security	Kaspersky	лицензионное
4	Microsoft Office 2016	Microsoft	лицензионное
5	Zoom	Zoom	лицензионное
6	Python 3.9	Python	Свободно распространяемое
7	Visual Studio Code Community	Microsoft	Свободно распространяемое
8	PyCharm Community	Jetbrains	Свободно распространяемое
9	Jupyter Laboratory	Project Jupyter	Свободно распространяемое

8. Обеспечение образовательного процесса для лиц с ограниченными возможностями здоровья и инвалидов

В ходе реализации дисциплины используются следующие дополнительные методы обучения, текущего контроля успеваемости и промежуточной аттестации обучающихся в зависимости от их индивидуальных особенностей:

- для слепых и слабовидящих: лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением; письменные задания выполняются на компьютере со специализированным программным обеспечением или могут быть заменены устным ответом; обеспечивается индивидуальное равномерное освещение не менее 300 люкс; для выполнения задания при необходимости предоставляется увеличивающее устройство; возможно также использование собственных увеличивающих устройств; письменные задания оформляются увеличенным шрифтом; экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

- для глухих и слабослышащих: лекции оформляются в виде электронного документа, либо предоставляется звукоусиливающая аппаратура индивидуального пользования; письменные задания выполняются на компьютере в письменной форме; экзамен и зачёт проводятся в письменной форме на компьютере; возможно проведение в форме тестирования.

- для лиц с нарушениями опорно-двигательного аппарата: лекции оформляются в виде электронного документа, доступного с помощью компьютера со специализированным программным обеспечением; письменные задания выполняются на компьютере со специализированным программным обеспечением; экзамен и зачёт проводятся в устной форме или выполняются в письменной форме на компьютере.

При необходимости предусматривается увеличение времени для подготовки ответа.

Процедура проведения промежуточной аттестации для обучающихся устанавливается с учётом их индивидуальных психофизических особенностей. Промежуточная аттестация может проводиться в несколько этапов.

При проведении процедуры оценивания результатов обучения предусматривается использование технических средств, необходимых в связи с индивидуальными особенностями обучающихся. Эти средства могут быть предоставлены университетом, или могут использоваться собственные технические средства.

Проведение процедуры оценивания результатов обучения допускается с использованием дистанционных образовательных технологий.

Обеспечивается доступ к информационным и библиографическим ресурсам в сети Интернет для каждого обучающегося в формах, адаптированных к ограничениям их здоровья и восприятия информации:

- для слепых и слабовидящих: в печатной форме увеличенным шрифтом, в форме электронного документа, в форме аудиофайла.

- для глухих и слабослышащих: в печатной форме, в форме электронного документа.

- для обучающихся с нарушениями опорно-двигательного аппарата: в печатной форме, в форме электронного документа, в форме аудиофайла.

Учебные аудитории для всех видов контактной и самостоятельной работы, научная библиотека и иные помещения для обучения оснащены специальным оборудованием и учебными местами с техническими средствами обучения:

- для слепых и слабовидящих: устройством для сканирования и чтения с камерой SARA SE; дисплеем Брайля PAC Mate 20; принтером Брайля EmBraille ViewPlus;

- для глухих и слабослышащих: автоматизированным рабочим местом для людей с нарушением слуха и слабослышащих; акустический усилитель и колонки;

- для обучающихся с нарушениями опорно-двигательного аппарата: передвижными, регулируемые эргономическими партами СИ-1; компьютерной техникой со специальным программным обеспечением.

9. Методические материалы

9.1 Планы семинарских/ практических/ лабораторных занятий

Тема 1.

- Основные понятия объектно-ориентированного программирования: наследование, полиморфизм, инкапсуляция. Основные принципы ООП: SOLID.
- Определение пользовательских классов в языке Python. Атрибуты и методы классов. Статические и динамические атрибуты. Методы экземпляра класса.

- Написание программ с использованием пользовательских классов и наследованием.

Тема 2.

- Методы экземпляра класса и магические методы экземпляра класса. Понятие перегрузки методов. Протокол итерации. Магические методы, связанные с протоколом итерации.

- Переопределение магических методов для итерирования.

Тема 3.

- Наследование в Python. Понятие родительского класса. Множественное наследование. Порядок разрешения методов: MRO.

- Альтернатива наследованию: композиция классов.
- Написание программ с использованием множественного наследования и подмешивания классов.

Тема 4.

- Понятие инкапсуляции. Интерфейс класса. Особенности реализации инкапсуляции в языке Python.

- Псевдозакрытые атрибуты и методы.
- Написание программ с использованием классов с инкапсуляцией.

Тема 5.

- Понятие дескриптора. Имплементация дескрипторов в языке Python. Реализация дескрипторов с помощью функций и классов. Property.

- Исключения: обработка исключений. Оператор try\except. Создание собственных исключений.

- Написание программ с перехватом исключений и использованием дескрипторов атрибутов класса.

Тема 6.

- Понятие декоратора. Синтаксис декоратора в Python. Closure.
- Методы класса и статические методы.
- Написание программ с использованием декораторов для создания статических методов, методов класса и дескрипторов. Написание фабрики декораторов.

Тема 7.

- Понятие метакласса. Создание пользовательских классов с использованием функции type(). Абстрактные классы и методы.

- Написание программ с использованием метаклассов и абстрактных классов.

9.2 Иные материалы

Все необходимые для обучения материалы публикуются по адресу <https://github.com/rsuh-python/> в соответствующих репозиториях.