



МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования
«Магнитогорский государственный технический университет им. Г.И. Носова»



УТВЕРЖДАЮ
Директор ИЭиАС
В.Р. Храмшин

26.01.2022 г.

РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ (МОДУЛЯ)

ПРОГРАММИРОВАНИЕ НА ЯЗЫКЕ PYTHON

Направление подготовки (специальность)
15.04.06 Мехатроника и робототехника

Направленность (профиль/специализация) программы
Искусственный интеллект в робототехнике

Уровень высшего образования - магистратура

Форма обучения
очная

Институт/ факультет	Институт энергетики и автоматизированных систем
Кафедра	Автоматизированного электропривода и мехатроники
Курс	1
Семестр	1

Магнитогорск
2022 год

Рабочая программа составлена на основе ФГОС ВО - магистратура по направлению подготовки 15.04.06 Мехатроника и робототехника (приказ Минобрнауки России от 14.08.2020 г. № 1023)

Рабочая программа рассмотрена и одобрена на заседании кафедры Автоматизированного электропривода и мехатроники

17.01.2022, протокол № 5

Зав. кафедрой А.А. Николаев

Рабочая программа одобрена методической комиссией ИЭиАС

26.01.2022 г. протокол № 5

Председатель В.Р. Храмшин

Рабочая программа составлена:

доцент кафедры АЭПиМ, канд. техн. наук О.С. Малахов

Рецензент:

зам. начальника ЦЭТЛ ПАО «ММК» по электроприводу, канд. техн. наук

А.Ю. Юдин



Лист актуализации рабочей программы

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2023 - 2024 учебном году на заседании кафедры Автоматизированного электропривода и

Протокол от ____ 20__ г. № ____
Зав. кафедрой _____ А.А. Николаев

Рабочая программа пересмотрена, обсуждена и одобрена для реализации в 2024 - 2025 учебном году на заседании кафедры Автоматизированного электропривода и

Протокол от ____ 20__ г. № ____
Зав. кафедрой _____ А.А. Николаев

1 Цели освоения дисциплины (модуля)

Целью курса является изучение основных понятий и принципов объектно-ориентированного программирования (ООП), алгоритмизации, а также

практических навыков программирования на языке высокого уровня Python.

2 Место дисциплины (модуля) в структуре образовательной программы

Дисциплина Программирование на языке Python входит в часть учебного плана формируемую участниками образовательных отношений образовательной программы.

Для изучения дисциплины необходимы знания (умения, владения), сформированные в результате изучения дисциплин/ практик:

Требования к «входным» знаниям, умениям, навыкам студента, необходимым при освоении данной дисциплины и приобретенным в результате освоения предшествующих дисциплин отсутствуют.

Знания (умения, владения), полученные при изучении данной дисциплины будут необходимы для изучения дисциплин/практик:

Интеллектуальные системы управления в мехатронике и робототехнике

Машинное обучение

3 Компетенции обучающегося, формируемые в результате освоения дисциплины (модуля) и планируемые результаты обучения

В результате освоения дисциплины (модуля) «Программирование на языке Python» обучающийся должен обладать следующими компетенциями:

Код индикатора	Индикатор достижения компетенции
ОПК-11	Способен организовывать разработку и применение алгоритмов и современных цифровых программных методов расчетов и проектирования отдельных устройств и подсистем мехатронных и робототехнических систем с использованием стандартных исполнительных и управляющих устройств, средств автоматики, измерительной и вычислительной техники в соответствии с техническим заданием, разрабатывать цифровые алгоритмы и программы управления робототехнических систем;
ОПК-11.1	Знает: Методы и программные средства автоматизированного проектирования нормативно-технической документации мехатронных и робототехнических систем; Основы высшей математики, алгоритмизации технологических процессов.; Порядок и способы разработки цифровых алгоритмов и программ при проектировании интеллектуальных модулей управления робототехническими и мехатронными системами; Методы построения алгоритмов, основы высшей математики и математической статистики
ОПК-11.2	Умеет: Применять программный инструментарий разработки технического и программного обеспечения мехатронных и робототехнических систем; Разрабатывать алгоритмы управления для робототехнических систем и реализовывать их в виде программного обеспечения; Применять методы искусственного интеллекта в разработке интеллектуальных алгоритмов управления робототехническими и мехатронными системами, а также выполнять их программную реализацию в процессе проектирования управляющих подсистем; Разрабатывать интеллектуальные модели и алгоритмы управления для мехатронных и робототехнических систем основываясь на экспериментальных и расчётных данных

ОПК-11.3	Имеет практический опыт: Владения методами и инструментами компьютерного проектирования мехатронных и робототехнических систем; Применение современных методов компьютерного проектирования цифровых систем с использованием элементов программируемой логики; Выполнения и организации разработки интеллектуальных алгоритмов управления и их программной реализации при проектировании подсистем управления робототехническими и мехатронными системами; Применение классических методов математической статистики и/или алгоритмов искусственного интеллекта для проектирования цифровых систем
----------	---

4. Структура, объём и содержание дисциплины (модуля)

Общая трудоемкость дисциплины составляет 6 зачетных единиц 216 акад. часов, в том числе:

- контактная работа – 64,9 акад. часов;
- аудиторная – 64 акад. часов;
- внеаудиторная – 0,9 акад. часов;
- самостоятельная работа – 151,1 акад. часов;
- в форме практической подготовки – 0 акад. час;

Форма аттестации - зачет с оценкой

Раздел/ тема дисциплины	Семестр	Аудиторная контактная работа (в акад. часах)			Самостоятельная работа студента	Вид самостоятельной работы	Форма текущего контроля успеваемости и промежуточной аттестации	Код компетенции
		Лек.	лаб. зан.	практ. зан.				
1. Основные понятия, синтаксис, стандарты и правила построения кода								
1.1 Введение, обзор курса. Язык программирования Python, начало работы	1	2		4	10	Подготовка к практическому занятию	Выполнение практического задания	ОПК-11.1, ОПК-11.2, ОПК-11.3
Итого по разделу		2		4	10			
2. Структуры данных, функции, методы								
2.1 Переменные и простые типы данных. Основные функции, ввод, чтение, обработка информации, взаимодействие с пользователем	1	2		4	10	Подготовка к практическому занятию	Выполнение практического задания	ОПК-11.1, ОПК-11.2, ОПК-11.3
2.2 Списки, кортежи, словари		2		6	10	Подготовка к практическому занятию	Выполнение практического задания	ОПК-11.1, ОПК-11.2, ОПК-11.3
Итого по разделу		4		10	20			
3. Операторы ветвления, циклы, исключения, модули, функции								
3.1 Условный оператор if. Логический тип данных. Циклы	1	2		14	27	Подготовка к практическому занятию	Выполнение практического задания	ОПК-11.1, ОПК-11.2, ОПК-11.3
Итого по разделу		2		14	27			
4. Введение в объектно-ориентированное программирование, принципы, структуры								
4.1 Объектно-ориентированная парадигма программирования. Классы	1	4		4	30	Подготовка к практическому занятию	Выполнение практического задания	ОПК-11.1, ОПК-11.2, ОПК-11.3
4.2 Понятие иерархии. Инкапсуляция, наследование, полиморфизм		2		8	30	Подготовка к практическому занятию	Выполнение практического задания	ОПК-11.1, ОПК-11.2, ОПК-11.3

Итого по разделу		6		12	60			
5. Библиотеки для работы с данными								
5.1 Python community. Наиболее распространённые библиотеки и их возможности	1	2		8	30	Подготовка к практическому занятию	Выполнение практического задания	ОПК-11.1, ОПК-11.2, ОПК-11.3
Итого по разделу		2		8	30			
6. Контроль								
6.1 Контроль	1					Подготовка к зачету		
Итого по разделу					4,1			
Итого за семестр		16		48	147		зао	
Итого по дисциплине		16		48	151,1		зачет с оценкой	

5 Образовательные технологии

Лекции проходят в традиционной форме и в форме лекций-консультаций. На лекциях-консультациях изложение нового материала сопровождается постановкой вопросов и дискуссией в поисках ответов на эти вопросы.

При выполнении практических заданий студенты учатся навыками написания программ, рассмотренных на лекционных занятиях.

Самостоятельная работа стимулирует студентов в процессе подготовки домашних заданий, при решении задач на практических занятиях, при подготовке к аттестации.

6 Учебно-методическое обеспечение самостоятельной работы обучающихся

Представлено в приложении 1.

7 Оценочные средства для проведения промежуточной аттестации

Представлены в приложении 2.

8 Учебно-методическое и информационное обеспечение дисциплины (модуля)

а) Основная литература:

Демидова, Л. А. Интеллектуальный анализ данных на языке Python : учебно-методическое пособие / Л. А. Демидова. — Москва : РТУ МИРЭА, 2021. — 92 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/218693> (дата обращения: 06.07.2022). — Режим доступа: для авториз. пользователей.

б) Дополнительная литература:

Демидова, Л. А. Кластерный анализ. Python : учебное пособие / Л. А. Демидова. — Москва : РТУ МИРЭА, 2022. — 103 с. — Текст : электронный // Лань : электронно-библиотечная система. — URL: <https://e.lanbook.com/book/240092> (дата обращения: 06.07.2022). — Режим доступа: для авториз. пользователей.

в) Методические указания:

1. Учебно-методическое пособие по выполнению семестрового задания по курсу "Программирование на языке Python"

г) Программное обеспечение и Интернет-ресурсы:

Программное обеспечение

Наименование ПО	№ договора	Срок действия лицензии
7Zip	свободно распространяемое ПО	бессрочно
JetBrains PyCharm Community Edition	свободно распространяемое ПО	бессрочно

Профессиональные базы данных и информационные справочные системы

Название курса	Ссылка
Электронная база периодических изданий East View Information Services, ООО «ИВИС»	https://dlib.eastview.com/
Национальная информационно-аналитическая система – Российский индекс научного цитирования (РИНЦ)	URL: https://elibrary.ru/project_risc.asp

9 Материально-техническое обеспечение дисциплины (модуля)

Материально-техническое обеспечение дисциплины включает:

Учебные аудитории для проведения занятий лекционного типа

Учебные аудитории для проведения групповых и индивидуальных консультаций, текущего контроля и промежуточной аттестации

Учебные аудитории для самостоятельной работы обучающихся

Учебно-методическое обеспечение самостоятельной работы обучающихся

По дисциплине «Программирование на языке Python» предусмотрена аудиторная и внеаудиторная самостоятельная работа обучающихся.

Аудиторная самостоятельная работа студентов предполагает ответы на вопросы на практических занятиях при защите работ.

Примерные вопросы для устного опроса и защиты практических работ и зачета:

1. Какие знаете принципы программирования?
2. Чем отличаются процедурная и объектов-ориентированная парадигмы программирования?
3. Какие основные принципы ООП (наследование, инкапсуляция, полиморфизм)?
4. Что такое множественное наследование?
5. Python - интерпретируемый язык или компилируемый?
6. Какие есть меняющиеся и постоянные типы данных?
7. Что такое область видимости переменных?
8. Что такое introspection?
9. Разница между `is` и `==`?
10. Разница между `__init__()` и `__new__()`?
11. В чем разница между потоками и процессами?
12. Какие есть виды импорта?
13. Что такое класс, итератор, генератор?
14. Что такое метакласс, переменная цикла?
15. В чем разница между `staticmethod` и `classmethod`?
16. Какие есть основные популярные пакеты (`requests`, `pytest`, etc)?
17. Что такое `lambda`-функции?
18. Что означает `*args`, `**kwargs` и как они используются?
19. Что такое `exceptions`, `<try-except>`?
20. Как использовать встроенные коллекции (`list`, `set`, `dictionary`)?

Примерные задания для практических работ и зачета:

1. Написать функцию `arithmetic`, принимающую 3 аргумента: первые 2 - числа, третий - операция, которая должна быть произведена над ними. Если третий аргумент `+`, сложить их; если `-`, то вычесть; `*` — умножить; `/` — разделить (первое на второе). В остальных случаях вернуть строку "Неизвестная операция".
2. Написать функцию `is_year_leap`, принимающую 1 аргумент — год, и возвращающую `True`, если год високосный, и `False` иначе.
3. Написать функцию `square`, принимающую 1 аргумент — сторону квадрата, и возвращающую 3 значения (с помощью кортежа): периметр квадрата, площадь квадрата и диагональ квадрата.

4. Написать функцию `season`, принимающую 1 аргумент — номер месяца (от 1 до 12), и возвращающую время года, которому этот месяц принадлежит (зима, весна, лето или осень).
5. Написать функцию `is_prime`, принимающую 1 аргумент — число от 0 до 1000, и возвращающую `True`, если оно простое, и `False` - иначе.
6. Написать функцию `date`, принимающую 3 аргумента — день, месяц и год. Вернуть `True`, если такая дата есть в нашем календаре, и `False` иначе.

а) Планируемые результаты обучения и оценочные средства для проведения промежуточной аттестации:

Код индикатора	Индикатор достижения компетенции	Оценочные средства
<i>ОПК-11: Способен организовывать разработку и применение алгоритмов и современных цифровых программных методов расчетов и проектирования отдельных устройств и подсистем мехатронных и робототехнических систем с использованием стандартных исполнительных и управляющих устройств, средств автоматики, измерительной и вычислительной техники в соответствии с техническим заданием, разрабатывать цифровые алгоритмы и программы управления робототехнических систем</i>		
ОПК-11.1	Знает: Методы и программные средства автоматизированного проектирования нормативно-технической документации мехатронных и робототехнических систем; Основы высшей математики, алгоритмизации технологических процессов.; Порядок и способы разработки цифровых алгоритмов и программ при проектировании интеллектуальных модулей управления робототехническими и мехатронными системами; Методы построения алгоритмов, основы высшей математики и математической статистики	Примерные вопросы для устного опроса и защиты практических работ и зачета: 1. Какие знаете принципы программирования? 2. Чем отличаются процедурная и объектов-ориентированная парадигмы программирования? 3. Какие основные принципы ООП (наследование, инкапсуляция, полиморфизм)? 4. Что такое множественное наследование? 5. Python - интерпретируемый язык или компилируемый? 6. Какие есть меняющиеся и постоянные типы данных? 7. Что такое область видимости переменных? 8. Что такое introspection? 9. Разница между is и ==? 10. Разница между __init__ () и __new__ ()? 11. В чем разница между потоками и процессами?
ОПК-11.2	Умеет: Применять программный инструментарий разработки технического и программного обеспечения мехатронных и робототехнических систем; Разрабатывать алгоритмы управления для робототехнических систем и реализовывать	Примерные вопросы для устного опроса и защиты практических работ и зачета: 12. Какие есть виды импорта? 13. Что такое класс, итератор, генератор? 14. Что такое метакласс, переменная цикла? 15. В чем разница между staticmethod и classmethod? 16. Какие есть основные популярные пакеты (requests, pytest, etc)? 17. Что такое lambda-функции? 18. Что означает *args, **kwargs и как они используются?

	их в виде программного обеспечения; Применять методы искусственного интеллекта в разработке интеллектуальных алгоритмов управления робототехническими и мехатронными системами, а также выполнять их программную реализацию в процессе проектирования управляющих подсистем; Разрабатывать интеллектуальные модели и алгоритмы управления для мехатронных и робототехнических систем основываясь на экспериментальных и расчётных данных	19. Что такое exceptions, <try-except>? 20. Как использовать встроенные коллекции (list, set, dictionary)?
ОПК-11.3	Имеет практический опыт: Владения методами и инструментами компьютерного проектирования мехатронных и робототехнических систем; Применение современных методов компьютерного проектирования цифровых систем с использованием элементов программируемой логики; Выполнения и организации разработки интеллектуальных алгоритмов управления и их программной реализации при проектировании подсистем управления робототехническими и мехатронными системами; Применение классических методов математической	Примерные задания для практических работ и зачета: 1. Написать функцию arithmetic, принимающую 3 аргумента: первые 2 - числа, третий - операция, которая должна быть произведена над ними. Если третий аргумент +, сложить их; если —, то вычесть; * — умножить; / — разделить (первое на второе). В остальных случаях вернуть строку "Неизвестная операция". 2. Написать функцию is_year_leap, принимающую 1 аргумент — год, и возвращающую True, если год високосный, и False иначе. 3. Написать функцию square, принимающую 1 аргумент — сторону квадрата, и возвращающую 3 значения (с помощью кортежа): периметр квадрата, площадь квадрата и диагональ квадрата. 4. Написать функцию season, принимающую 1 аргумент — номер месяца (от 1 до 12), и возвращающую время года, которому этот месяц принадлежит (зима, весна, лето или осень). 5. Написать функцию is_prime, принимающую 1 аргумент — число от 0 до 1000, и возвращающую True, если оно простое, и False - иначе. 6. Написать функцию date, принимающую 3 аргумента — день, месяц и год. Вернуть True, если такая дата есть в нашем календаре, и False

	статистики и/или алгоритмов искусственного интеллекта для проектирования цифровых систем	иначе.
--	--	--------

б) Порядок проведения промежуточной аттестации, показатели и критерии оценивания:

Промежуточная аттестация по дисциплине «Программирование на языке Python» включает теоретические вопросы, позволяющие оценить уровень усвоения обучающимися знаний, и практические задания, выявляющие степень сформированности умений и владений.

Показатели и критерии оценивания практических работ и зачета:

– на оценку **«отлично»** (5 баллов) – обучающийся демонстрирует высокий уровень сформированности компетенций, всестороннее, систематическое и глубокое знание учебного материала, свободно выполняет практические задания, свободно оперирует знаниями, умениями, применяет их в ситуациях повышенной сложности.

– на оценку **«хорошо»** (4 балла) – обучающийся демонстрирует средний уровень сформированности компетенций: основные знания, умения освоены, но допускаются незначительные ошибки, неточности, затруднения при аналитических операциях, переносе знаний и умений на новые, нестандартные ситуации.

– на оценку **«удовлетворительно»** (3 балла) – обучающийся демонстрирует пороговый уровень сформированности компетенций: в ходе контрольных мероприятий допускаются ошибки, проявляется отсутствие отдельных знаний, умений, навыков, обучающийся испытывает значительные затруднения при оперировании знаниями и умениями при их переносе на новые ситуации.

– на оценку **«неудовлетворительно»** (2 балла) – обучающийся демонстрирует знания не более 20% теоретического материала, допускает существенные ошибки, не может показать интеллектуальные навыки решения простых задач.

– на оценку **«неудовлетворительно»** (1 балл) – обучающийся не может показать знания на уровне воспроизведения и объяснения информации, не может показать интеллектуальные навыки решения простых задач.