

**Автономная некоммерческая организация  
дополнительного профессионального образования  
«Центральный многопрофильный институт  
профессиональной переподготовки и повышения квалификации»  
АНО ДПО «ЦМИ»**



**УТВЕРЖДАЮ**  
Директор АНО ДПО «ЦМИ»  
А.В. Гриднева  
«18» апреля 2023 г.

**Дополнительная общеобразовательная  
общеразвивающая программа  
«Программирование БПЛА на Python: от теории к практике»**

**144 часа**

*Направленность: техническая  
уровень сложности «Продвинутый»*

*144 часа*

**Общие данные о Дополнительной общеобразовательной программе  
«Программирование БПЛА на Python: от теории к практике»**

**Об организации**

<b>Наименование поля</b>	<b>Допустимые значения поля</b>	<b>Значение поля</b>
ИНН организации, осуществляющей образовательную деятельность	10 арабских цифр	2632108764
Наименование организации	строка	АВТОНОМНАЯ НЕКОММЕРЧЕСКАЯ ОРГАНИЗАЦИЯ ДОПОЛНИТЕЛЬНОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ «ЦЕНТРАЛЬНЫЙ МНОГОПРОФИЛЬНЫЙ ИНСТИТУТ ПРОФЕССИОНАЛЬНОЙ ПЕРЕПОДГОТОВКИ И ПОВЫШЕНИЯ КВАЛИФИКАЦИИ»
Логотип организации	изображение в формате jpeg разрешением не менее 100x100 пиксель	
Ссылка на логотип организации	URL на изображение, находящееся в сети интернет	<a href="https://drive.google.com/file/d/1nobPZvgUGUtrHOVXT3yDIPdLZHXTB7nc/view?usp=drive_link">https://drive.google.com/file/d/1nobPZvgUGUtrHOVXT3yDIPdLZHXTB7nc/view?usp=drive_link</a>
Контакты ответственного за программу (с	строка от 5 до 255 символов	Гончарова Ирина Сергеевна

указанием фамилии, имени, отчества).		
Контакты ответственного за программу. Должность	строка от 5 до 255 символов	РОП
Контакты ответственного за программу. Телефон	Формат +7(XXX)XXXXX XX	+7(929)6469339
Контакты ответственного за программу. E- mail	строка	umo_sk@mail.ru

### Информация о программе

Наименование поля	Допустимые значения поля	Значение поля (примеры)
Название программы (курса)	строка	Программирование БПЛА на Python: от теории к практике
Описание программы	строка не менее 1000 не более 5000 символов	<p>Дополнительная общеобразовательная программа «Программирование БПЛА на Python: от теории к практике» является неотъемлемой частью образовательной программы автономной некоммерческой организации дополнительного профессионального образования «Центральный многопрофильный институт профессиональной переподготовки и повышения квалификации» и дает возможность каждому ребенку получать дополнительное образование исходя из его интересов, склонностей и способностей, образовательных потребностей, осуществляемых за пределами федеральных государственных образовательных стандартов и федеральных государственных требований. По своему функциональному назначению программа является общеразвивающей и направлена на удовлетворение потребностей, обучающихся в интеллектуальном, нравственном совершенствовании, в организации их свободного времени.</p> <p>Направленность общеобразовательной программы «Программирование БПЛА на Python: от теории к практике» техническая.</p>

<p>Аннотация (для размещения на маркетплейсе, понятное и привлекательное для Потенциальных получателей поддержки, включающее полное и содержательное описание Дополнительной общеобразовательной программы:</p> <p>1) краткое описание Дополнительной общеобразовательной программы;</p> <p>2) описание требований и рекомендаций для обучения по образовательной Дополнительной общеобразовательной программе;</p> <p>3) краткое описание результатов обучения в свободной форме, включая описание практикоориентированного характера Дополнительной общеобразовательной программы)</p>	<p>строка до 1000 символов</p>	<p>Дополнительная общеобразовательная программа «Программирование беспилотных летательных аппаратов на языке Python: от теории к практике» предназначен для тех, кто хочет познакомиться с основами программирования беспилотных летательных аппаратов на языке Python. В курсе будут рассмотрены как теоретические аспекты, так и практические задачи, позволяющие углубить знания в данной области.</p> <p>Курс начинается с введения в понятия беспилотных летательных аппаратов и их применения в различных отраслях. Затем следует обзор основных компонентов беспилотных летательных аппаратов, таких как бортовые компьютеры, сенсоры, системы навигации и т.д.</p> <p>После этого курс переходит к изучению основ программирования на языке Python. Будут рассмотрены основные конструкции языка, типы данных, функции и объектно-ориентированное программирование. Также будет дан обзор библиотек, которые используются при программировании беспилотных летательных аппаратов.</p> <p>Далее курс переходит к конкретным практическим задачам. Будет рассмотрено, как программировать автономный полет беспилотных летательных аппаратов, в том числе задачи по навигации, управлению, сбору и анализу данных. Также будут рассмотрены вопросы безопасности и ограничений, связанных с программированием беспилотных летательных аппаратов.</p> <p>В конце курса обучающимся будет предложена возможность самостоятельно реализовать проект, используя полученные знания. Это может быть, например, разработка программного обеспечения для управления беспилотным летательным аппаратом или алгоритм для сбора и анализа данных.</p>
<p>Цель программы</p>	<p>строка не менее 100 символов</p>	<p>Основная цель программы «Программирование беспилотных летательных аппаратов на языке Python: от теории к практике» - обучить основам программирования беспилотных летательных аппаратов на языке Python.</p> <p>В рамках программы учащиеся ознакомятся с теоретическими аспектами беспилотных летательных аппаратов, компонентами, необходимыми для их работы, а также с основами языка программирования Python.</p>

		<p>Кроме того, учащиеся изучат практические аспекты программирования беспилотных летательных аппаратов, включая автономный полет, навигацию, управление и сбор и анализ данных.</p> <p>В результате прохождения программы учащиеся смогут применять полученные знания для решения практических задач в области беспилотных летательных аппаратов, а также разрабатывать собственные проекты в этой области.</p>
Актуальность	строка не менее 500 символов	Актуальность программы обусловлена потребностью общества в технически грамотных специалистах и полностью отвечает социальному заказу по подготовке квалифицированных кадров в области программирования, а также высоким интересом подростков к IT- сфере. Важнейшей характеристикой подрастающего поколения является активность в информационном пространстве, интернет- коммуникации.
Дополнительная информация	строка	Обучающиеся приобретают необходимые навыки по программированию цифровых продуктов, что им помогает определиться с профессиональной сферой деятельности на будущее.
Формат обучения	значение из: очная форма без применения дистанционных образовательных технологий; очная форма с применением дистанционных образовательных технологий, в том числе, с применением средств электронного обучения	Очная форма без применения дистанционных образовательных технологий; Очная форма с применением дистанционных образовательных технологий, в том числе, с применением средств электронного обучения
Уровень сложности	значение из: «Начальный» «Базовый» «Продвинутый»	Продвинутый

Срок освоения образовательной программы	строка, значение в ак.ч.	144 ак.ч.
Объем каждого модуля в ак.ч.	целое число	36
Объем часов в неделю в ак.ч.	целое число	4
Количество занятий	целое число	58
Направленность программы	строка	Техническая
Язык программирования	строка	Python
Дополнительная общеобразовательная программа не представлена для участия в иных федеральных проектах, направленных на дополнительное образование граждан, кроме федерального проекта «Развитие кадрового потенциала ИТ- отрасли»	строка, значения: «Не представлена»	Не представлена
Дополнительная общеобразовательная программа не была реализована до начала отбора и/или не реализовывается в период отбора на безвозмездной основе	строка, значения «Не реализована»	Не реализована
Категория обучающихся по программе	строка не менее 10 символов	Учащиеся 9 класса, Учащиеся 10 класса, Учащиеся 11 класса, Обучающиеся по программам среднего профессионального образования
Описание планируемых результатов обучения	строка не менее 10 символов	Сформировать у обучающихся познавательный интерес к процессу сайтостроения; Сформировать знания и умения языка программирования Python,

		Сформировать умение применять программные средства для решения задач из различных предметных областей способствует развитию логического и комбинаторного мышления
Ссылка на лендинг Образовательной программы	строка не менее 10 символов	<a href="https://edu-sigma.ru">Курс «Программирование БПЛА на Python: от теории к практике» — онлайн- обучение бесплатно (edu- sigma.ru)</a>
Ссылка на LMS	строка не менее 10 символов	<a href="https://odin.study/ru/">https://odin.study/ru/</a>
Страница обучения на курсе	строка не менее 10 символов	<a href="https://odin.study">Программирование БПЛА на Python: от теории к практике (odin.study)</a>

## Аттестация

<b>Промежуточная аттестация</b>		
Количество академических часов	строка не менее 10 символов	4 часа текущая работа, включающая выполнение 4 практических задания
Формы контроля	строка не менее 10 символов	Для успешного завершения обучения по курсу необходимо, чтобы 50 % заданий в рамках текущей работы были зачтены (оценены преподавателем на 3, 4 или 5 баллов).
Диагностические инструменты	строка не менее 10 символов	Оценка полноты выполнения работы. Оценка способности оперировать полученными знаниями и умениями при решении практической задачи. Проверка результата запуска программного решения на языке Python.
Показатели и критерии оценивания	строка не менее 10 символов	<p><b>Шкала оценивания:</b>                      Нижнее значение - оценка «2».                      Верхнее значение - оценка «5».                      Минимальный проходной балл для успешной сдачи - оценка «3».</p> <p>При оценке надо руководствоваться следующими критериями:</p> <ol style="list-style-type: none"> <li>1) полнота и правильность выполнения заданий;</li> <li>2) степень осознанности, понимания изученного;</li> <li>3) оформление ответа.</li> </ol> <p><b>Оценка «5» ставится, если:</b></p> <ol style="list-style-type: none"> <li>1) Практическое задание выполнено без ошибок, полно изложен изученный материал;</li> <li>2) учащийся понимает материал, может обосновать свои суждения и действия, хорошо применяет знания на практике, приводит необходимые примеры не только по учебному материалу, но и самостоятельно составленные;</li> </ol> <p><b>Оценка «4» ставится, если</b> ученик выполнил практическое задание, удовлетворяющее тем же требованиям, что и для оценки «5», но допускает 1—2 ошибки, которые сам же исправляет, и 1—2 недочета в последовательности излагаемого материала.</p> <p><b>Оценка «3» ставится, если</b> ученик освоил знания и понимает основы практического задания, но:</p>

		<p>1) подготовил материал неполно и допускает неточности в последовательности выполнения заданий;  2) не может обосновать свои суждения и привести свои примеры;  3) допускает ошибки в оформлении.</p> <p><b>Оценка «2» ставится, если</b> ученик обнаруживает незнание большей части изучаемого материала, допускает ошибки в написании кода, беспорядочно и неуверенно излагает материал. Оценка «2» отмечает такие недостатки в подготовке ученика, которые являются серьезным препятствием к успешному овладению последующим материалом.</p>
Примеры заданий	строка не менее 10 символов	<p><b>Модуль 1.</b> Практическое задание № 1. Основы программирования беспилотных летательных аппаратов на языке Python.</p> <p>Шаг 1. Определить координаты начальной точки, координаты конечной точки и скорость полета беспилотного летательного аппарата.</p> <p>Шаг 2. Написать функцию, которая будет рассчитывать расстояние между двумя точками на плоскости по их координатам.</p> <p>Шаг 3. Написать функцию, которая будет рассчитывать время полета беспилотного летательного аппарата до конечной точки по расстоянию и скорости полета.</p> <p>Шаг 4. Написать функцию, которая будет рассчитывать направление полета беспилотного летательного аппарата по координатам начальной и конечной точек.</p> <p>Шаг 5. Написать функцию, которая будет управлять движением беспилотного летательного аппарата по прямой до достижения конечной точки.</p> <p>Шаг 6. Написать цикл, который будет вызывать функцию управления движением беспилотного летательного аппарата до тех пор, пока он не достигнет конечной точки.</p> <p>Шаг 7. Вывести сообщение об успешном завершении полета.</p> <p>Шаг 8. Проверить программу на работоспособность.</p> <p>Шаг 9. Протестировать программу на нескольких наборах начальных и конечных координат.</p> <p>Шаг 10. Оптимизировать программу, улучшив ее скорость и точность.</p> <p>Шаг 11. Загрузите файл в Odin.</p> <p><b>Модуль 2.</b> Практическое задание № 2. Продвинутые техники программирования беспилотных летательных аппаратов на языке Python</p> <p>Шаг 1. Создайте новый проект в вашей среде разработки, назовите его «Детектор объектов».</p> <p>Шаг 2. Загрузите изображение с объектом, который вы хотите обнаружить, например, изображение машины.</p>

	<p>Шаг 3. Используя библиотеку OpenCV для Python, напишите код, который будет обнаруживать машину на изображении. Воспользуйтесь, например, алгоритмом каскадных классификаторов Хаара (Haar Cascade).</p> <p>Шаг 4. Добавьте код для вывода прямоугольной рамки вокруг найденного объекта (машины).</p> <p>Шаг 5. Попробуйте запустить ваш код на других изображениях с машинами и проверьте, работает ли ваш детектор объектов правильно.</p> <p>Шаг 6: Загрузите файл в Odin.</p> <p><b>Модуль 3.</b> Практическое задание № 3 Расширенные технологии и тенденции программирования беспилотных летательных аппаратов на языке Python.</p> <p>Шаг 1. Создайте класс для беспилотного летательного аппарата, который будет содержать основные характеристики, такие как высота, скорость, угол наклона, координаты, дистанция и т.д.</p> <p>Шаг 2. Используйте библиотеку OpenCV для обработки изображений. Например, реализуйте распознавание объектов на изображениях, используя нейронные сети.</p> <p>Шаг 3. Разработайте алгоритм управления беспилотным летательным аппаратом с помощью машинного обучения. Например, можно использовать алгоритмы глубокого обучения для обучения аппарата выполнять определенные действия в зависимости от условий и среды.</p> <p>Шаг 4. Напишите код для управления беспилотным летательным аппаратом, используя библиотеку для управления полетом, такую как DroneKit. Шаг 5. Добавьте функциональность для получения и отправки данных, управления двигателем и т.д.</p> <p>Шаг 6. Разработайте сценарий для тестирования беспилотного летательного аппарата. Например, можно создать сценарий для облета заданных объектов, съемки видео с высоты и т.д. Включите в сценарий проверку на ошибки и обработку исключительных ситуаций.</p> <p>Шаг 7. Загрузите файл в Odin.</p> <p><b>Модуль 4.</b> Практическое задание № 4. Практические примеры разработки беспилотных летательных аппаратов на языке Python.</p> <p>Шаг 1. Создайте класс для беспилотного летательного аппарата, который будет содержать основные характеристики, такие как высота, скорость, угол наклона, координаты, дистанция и т.д.</p> <p>Шаг 2. Напишите функцию для управления беспилотным летательным аппаратом. Функция должна использовать библиотеку для управления полетом, такую как DroneKit, и добавлять функциональность для получения и отправки данных, управления двигателем и т.д.</p> <p>Шаг 3. Разработайте сценарий для тестирования беспилотного летательного аппарата. Например, можно создать сценарий для облета заданных объектов, съемки видео с высоты и т.д. Включите в сценарий проверку на ошибки и обработку исключительных ситуаций.</p>
--	---

		<p>Шаг 4. Используйте библиотеку OpenCV для обработки изображений. Например, реализуйте распознавание объектов на изображениях, используя нейронные сети.</p> <p>Шаг 5. Разработайте алгоритм управления беспилотным летательным аппаратом с помощью машинного обучения. Например, можно использовать алгоритмы глубокого обучения для обучения аппарата выполнять определенные действия в зависимости от условий и среды.</p> <p>Шаг 6. Загрузите файл в Odin.</p>
Шкала оценивания, нижнее значение	строка не менее 10 символов	Шкала оценивания: Нижнее значение - оценка «2».
Шкала оценивания, верхнее значение	строка не менее 10 символов	Шкала оценивания: Верхнее значение - оценка «5».
Шкала оценивания, минимальный проходной балл	строка не менее 10 символов	Шкала оценивания: Минимальный проходной балл для успешной сдачи - оценка «3».

## Преподаватели

<b>ФИО</b>	<b>Наименование основного места работы</b>	<b>Должность</b>	<b>Высшее образование или среднее профессиональное образование по направлению «Образование и педагогические науки»</b>	<b>Высшее образование или среднее профессиона льное образование по иному направлени ю соответству ющим направленно сти ДОП</b>	<b>Ссылка на веб- страницы с портфолио</b>	<b>Информация о курсах повышения квалификац ии по профилю преподаваем ой дисциплины (за последние 3 года)</b>	<b>Пройдена промежуточн ая аттестация не менее чем за два года обучения по образователь ным программам высшего образования по специальнос тям и направления м подготовки, соответствую щим направленно сти ДОП</b>	<b>Отметка о полученно м согласи на обработку персональн ых данных</b>
<b>строка от 2 до 100 символов</b>	<b>строка от 2 до 255 символов.</b>	<b>строка от 2 до 255 символов</b>	<b>да/нет</b>	<b>да/нет</b>	<b>строка</b>		<b>да/нет</b>	<b>да/нет</b>
Мезенцев Роман Александрович	Fogstream	Преподава- тель курсов программи- рования	нет	да	<a href="https://docs.google.com/document/d/1hurZV7s5-iYNOZzzhhUK">https://docs.google.com/document/d/1hurZV7s5-iYNOZzzhhUK</a>	нет	нет	да

					<a href="#">zwI4or-TD1i/edit</a>			
Назарцев Максим Сергеевич	ГБПОУ “Железноводский художественно-строительный техникум имени казачьего генерала В.П. Бондарева”	Преподаватель специальных дисциплин	да	да	<a href="https://docs.google.com/document/d/1yvdxHYOctGpLw_zSCsaZy1FUc3KgSaGw/edit#">https://docs.google.com/document/d/1yvdxHYOctGpLw_zSCsaZy1FUc3KgSaGw/edit#</a>	Повышение квалификации: ГБПОУ ГТМАУ «Комплекс технических воздействий по поддержанию автотранспортных средств в исправном состоянии (на базе ООО «Специализированный застройщик «Третий РИМ»)), 72 часа, май 2021 год	нет	да

						<p>Повышение квалификац ии: ГБПОУ СРМК «Практика и методика реализации образователь ных программ СПО с учетом спецификац ии стандартов Ворлдскилл с по компетенции «Ремонт и обслуживан ие легковых автомобилей », 76 часов, май 2020 год</p> <p>Сертификат</p>		
--	--	--	--	--	--	---	--	--

						эксперта -мастера Ворлдскилл с по компетенции «Ремонт и обслуживан ие легковых автомобилей » (на 3 года), июль 2020 год		
--	--	--	--	--	--	--	--	--

## **I. ПОЯСНИТЕЛЬНАЯ ЗАПИСКА**

Дополнительная общеобразовательная программа «Программирование БПЛА на Python: от теории к практике» реализуется Автономной некоммерческой организации дополнительного профессионального образования «Центральный многопрофильный институт профессиональной переподготовки и повышения квалификации и дает возможность каждому ребенку получать дополнительное образование исходя из его интересов, склонностей и способностей, образовательных потребностей, осуществляемых за пределами федеральных государственных образовательных стандартов и федеральных государственных требований.

По своему функциональному назначению программа является общеразвивающей и направлена на удовлетворение потребностей, обучающихся в интеллектуальном, нравственном совершенствовании, в организации их свободного времени.

Направленность общеобразовательной программы «Программирование БПЛА на Python: от теории к практике» техническая, так как она ориентирована на изучение основных графических компьютерных программ Python в рамках их широкого использования, а также специальных профессиональных возможностей.

### **Актуальность и педагогическая целесообразность программы**

Актуальность программы обусловлена потребностью общества в технически грамотных специалистах и полностью отвечает социальному заказу по подготовке квалифицированных кадров в области программирования, а также высоким интересом подростков к IT- сфере. Важнейшей характеристикой подрастающего поколения является активность в информационном пространстве, интернет- коммуникации.

Обучающиеся приобретают необходимые навыки по программированию цифровых продуктов, что им помогает определиться с профессиональной сферой деятельности на будущее.

Программа «Программирование БПЛА на Python: от теории к практике» модифицированная. Она составлена на основе типовых программ.

Знания, умения и практические навыки решения актуальных задач, полученные на занятиях, готовят обучающихся к самостоятельной проектно- исследовательской деятельности с применением современных технологий.

### **Цель и основные задачи программы**

Целью освоения языков программирования является изучение методов синтеза нейронных сетей и их практического применения.

### **Требования к уровню знаний, полученных в результате обучения.**

#### **Учащиеся должны знать:**

- основные типы алгоритмов, иметь представление о структуре программы, основы программирования на языках высокого уровня;
- базовые алгоритмические конструкции;

- содержание этапов разработки программы: алгоритмизация- кодирование- отладка тестирование;
- дополнительные возможности языка для выражения различных алгоритмических ситуаций; алгоритмы и программы на языке Python решения нестандартных задач, задач повышенной сложности в математической области; исходные данные и результаты, как строить алгоритмы методом последовательного уточнения (сверху вниз), изображать эти алгоритмы в виде блок- схем; дополнительные средства языка Python; основы постановки задач в области информационных систем.

#### **Учащиеся должны уметь:**

- записывать основные языки программирования для решения задач из области математики;
- использовать основные алгоритмические приемы при решении математических задач;
- решать нестандартные задачи и задачи повышенной сложности; анализировать текст чужих программ, находить в них неточности, оптимизировать алгоритм, создавать собственные варианты решения.

#### **Основные задачи:**

##### ***Обучающие:***

- познакомить учащихся с Python»;
- познакомить с понятиями алгоритма, вычислимой функции, языка программирования;
- научить читать и составлять блок- схемы;
- сформировать навыки выполнения технологической цепочки разработки программ средствами Python;  
изучить основные конструкции языка программирования Python, позволяющие работать с простыми и составными типами данных (строками, списками , кортежами, словарями, множествами);
- научить применять функции при написании программ на языке программирования.

##### ***Развивающие:***

- сформировать у учащихся познавательный интерес к процессу программирования;
- развить воображение, конструкторское мышление учащихся;
- сформировать видение возможностей использования, приобретенного на занятиях в учебной и повседневной деятельности;
- расширить знания по предмету и научить применять программные средства для решения задач из различных предметных областей;
- развить логическое и комбинаторное мышление.

##### ***Воспитательные:***

- воспитывать уважение и соблюдение авторских прав к работам окружающих;
- формировать положительный психологический климат в среде обучающихся.
- привить навыки культуры общения, деловых качеств, таких как ответственность, самостоятельность, активность, аккуратность.

## **Педагогические принципы, определяющие теоретические подходы к построению образовательного процесса**

Реализация программы «Программирование БПЛА на Python: от теории к практике» основывается на принципах научности, последовательности, системности, связи теории с практикой, доступности.

**Принцип доступности и понятности материала.** Курс должен быть доступен для всех учащихся, вне зависимости от их предыдущего опыта в области программирования и беспилотных летательных аппаратов. Материал должен быть представлен в понятной форме и объяснен в доступной форме.

**Принцип связывания теории с практикой.** Весь материал курса должен иметь практическое применение, чтобы учащиеся могли лучше понимать принципы и технологии, используемые в беспилотных летательных аппаратах.

**Принцип индивидуализации обучения.** Курс должен быть адаптирован под индивидуальные потребности каждого учащегося, учитывая их уровень знаний и умений, а также интересы и цели в обучении.

**Принцип активности учащегося.** Курс должен стимулировать активное участие в процессе обучения, включая решение задач, выполнение практических заданий и проектной работы.

**Принцип развития критического мышления.** Курс должен развивать у учащихся критическое мышление и способность к анализу и оценке информации в области программирования беспилотных летательных аппаратов.

**Принцип продуктивного сотрудничества.** Курс должен способствовать развитию навыков командной работы и сотрудничества, что позволит учащимся более эффективно работать в командах при проектировании и программировании беспилотных летательных аппаратов.

**Предполагаемые результаты** по окончанию реализации программы учащийся будет:

1. Знать основные алгоритмические конструкции программ.
2. Уметь писать на языке Python, простейшие программы, связанные с числовыми и символьными данными.
3. Уметь применять свои знания на практике при решении технических (математических задач).

## **Основные характеристики образовательного процесса**

Уровень освоения программы **продвинутый**, что предполагает освоение обучающимися специализированных знаний, обеспечение трансляции общей и целостной картины тематического содержания программы.

**Форма обучения:** очная форма с применением дистанционных образовательных технологий, в том числе, с применением средств электронного обучения; очная форма без применения дистанционных образовательных технологий;

**Срок реализации программы** - 144 ак. часа.

**Режим работы:** занятия проводятся 2- 3 раза в неделю по 2 академических часа, длительность одного академического часа – 45 минут.

**Объем учебных часов по программе** 144 часа.

**Ожидаемые результаты освоения программы**

Овладение предметными знаниями и умениями

*После завершения обучения по программе у обучающихся будут сформированы:*

- глубокое понимание основных концепций и принципов беспилотных летательных аппаратов и умение применять их на практике;
- умение работать с языком программирования Python, в том числе использовать основные библиотеки и инструменты для программирования беспилотных летательных аппаратов;
- навыки проектирования и разработки беспилотных летательных аппаратов с помощью языка программирования Python;
- умение выполнять тестирование, отладку и оптимизацию программ для беспилотных летательных аппаратов;
- опыт работы в команде и умение решать задачи, связанные с программированием беспилотных летательных аппаратов в коллективе;
- готовность к дальнейшему самостоятельному изучению и развитию в области программирования беспилотных летательных аппаратов на языке Python.

Данная программа адаптирована для занятий с учащимися среднего и старшего возраста, с различной степенью подготовки, включает теоретические и практические задания.

**Входные требования к обучающимся:** тестирование - базовое владение персональным компьютером.

**Для занятий** - наличие ПК, камеры, микрофона, стабильного доступа в интернет.

Вступительное тестирование должно отвечать следующим функциональным характеристикам: иметь не менее 25 вопросов. Вступительное тестирование должно состоять из мотивационных вопросов и заданий в области математики/информатики/программирования для определения уровня обучения. Вступительное тестирование должно оценивать мотивацию, предрасположенность и знания школьника для профессионального развития в сфере информационных технологий и получения соответствующего обучения.

**Календарный учебный график** формируется непосредственно при реализации программы общеобразовательной программы «Программирование БПЛА на Python: от теории к практике». Календарный учебный график представлен в форме расписания занятий при наборе группы на обучение.

## СОДЕРЖАНИЕ ОБЩЕРАЗВИВАЮЩЕЙ ПРОГРАММЫ

### СОДЕРЖАНИЕ УЧЕБНОГО (ТЕМАТИЧЕСКОГО) ПЛАНА

«Программирование БПЛА на Python: от теории к практике»

№	Наименование модулей (курсов)	Общая трудоемкость, ч.	Всего контактных часов		Контактные часы		СРС, ч.	Формы контроля
			Общее количество Лекций и ПЗ	ДЗ	Лекции	Практические занятия		
I	<b>Модуль 1: Продвинутое программирование и управление беспилотными летательными аппаратами с использованием Python.</b>	36	30	1	9	21	6	Зачет
1.1	Тема 1.1 Алгоритмы планирования траекторий для беспилотных летательных аппаратов.	1	1		1			
1.2	Тема 1.2 Реализация алгоритмов машинного обучения для автономного управления дронами	1	1		1			
1.3	Тема 1.3 Работа с компьютерным зрением и обработка изображений в контексте беспилотных летательных аппаратов	1	1		1			
1.4	Тема 1.4 Разработка системы оптической стабилизации камеры на дроне	7	6		1	5	1	
1.5	Тема 1.5 Использование датчиков и датчиков-обнаружителей (например, GPS, инфракрасные сенсоры, радары) для автономной навигации и	7	6		1	5	1	

	избегания препятствий							
1.6	Тема 1.6 Реализация системы распознавания и классификации объектов с помощью машинного обучения на дроне	4	3		1	2	1	
1.7	Тема 1.7 Оптимизация производительности беспилотных летательных аппаратов с использованием асинхронного программирования и многопоточности	5	4		1	3	1	
1.8	Тема 1.8 Разработка системы планирования миссий для беспилотных летательных аппаратов с учетом ограничений и целей	5	4		1	3	1	
1.9	Тема 1.9 Интеграция беспилотных летательных аппаратов с облачными сервисами для передачи данных, хранения информации и управления дронами из удаленного местоположения	5	4	1	1	3	1	<b>Зачет</b>
<b>2</b>	<b>Модуль 2: Продвинутое программирование беспилотных летательных аппаратов на языке Python</b>	<b>36</b>	<b>30</b>	<b>1</b>	<b>9</b>	<b>21</b>	<b>6</b>	<b>Зачет</b>
2.1	Тема 2.1 Обзор продвинутых технологий беспилотных летательных аппаратов	2	2		1	1		
2.2	Тема 2.2 Работа с датчиками и сенсорами беспилотных летательных аппаратов	3	3		1	2		
2.3	Тема 2.3 Обработка видео и изображений с помощью Python для беспилотных летательных	4	3		1	2	1	

	аппаратов							
2.4	Тема 2.4 Работа с базами данных для хранения данных беспилотных летательных аппаратов	4	3		1	2	1	
2.5	Тема 2.5 Разработка алгоритмов и программ для автоматического пилотирования беспилотных летательных аппаратов	4	3		1	2	1	
2.6	Тема 2.6 Использование машинного обучения и нейронных сетей в беспилотных летательных аппаратах	4	4		1	3		
2.7	Тема 2.7 Разработка и настройка систем управления беспилотными летательными аппаратами	3	3		1	2		
2.8	Тема 2.8 Создание пользовательских интерфейсов для управления беспилотными летательными аппаратами	4	3		1	2	1	
2.9	Тема 2.9 Проект: Разработка программы для автоматического пилотирования беспилотного летательного аппарата на Python	4	3	1	1	2	1	
2.10	Тема 2.10 Проект: Разработка системы управления беспилотным летательным аппаратом на Python	4	3			3	1	<b>Зачет</b>
<b>3</b>	<b>Модуль 3: Расширенные технологии и тенденции программирования беспилотных летательных аппаратов на языке Python</b>	<b>36</b>	<b>30</b>	<b>1</b>	<b>10</b>	<b>20</b>	<b>6</b>	<b>Зачет</b>

3.1	Тема 3.1 Разработка программного обеспечения для беспилотных летательных аппаратов на основе архитектур ROS и MAVLink	4	3		1	2	1	
3.2	Тема 3.2 Разработка и использование дополнительных средств связи между беспилотными летательными аппаратами и главным управляющим центром	4	3		1	2	1	
3.3	Тема 3.3 Разработка и применение технологий машинного обучения и искусственного интеллекта для беспилотных летательных аппаратов	4	3		1	2	1	
3.4	Тема 3.4 Интеграция различных датчиков и устройств для расширения функциональности беспилотных летательных аппаратов	4	3	1	1	2	1	<b>Зачет</b>
3.5	Тема 3.5 Применение компьютерного зрения и обработки изображений для улучшения функциональности беспилотных летательных аппаратов	3	3		1	2		
3.6	Тема 3.6 Разработка и применение системы навигации и автоматического пилотирования на основе датчиков GPS и инерциальной навигации	4	4		1	3		
3.7	Тема 3.7 Разработка системы обнаружения и предотвращения столкновений для беспилотных летательных аппаратов	4	4		1	3		

3.8	Тема 3.8 Проект: Разработка приложения для управления беспилотным летательным аппаратом на основе архитектуры ROS и MAVLink	4	3		1	2	1	
3.9	Тема 3.9 Проект: Разработка системы обнаружения и предотвращения столкновений для беспилотных летательных аппаратов	5	4		2	2	1	
<b>4</b>	<b>Модуль 4: Практические примеры разработки беспилотных летательных аппаратов на языке Python</b>	<b>36</b>	<b>30</b>	<b>3</b>	<b>9</b>	<b>21</b>	<b>6</b>	<b>Зачет</b>
4.1	Тема 4.1 Основы разработки беспилотных летательных аппаратов на языке Python	5	4		1	3	1	
4.2	Тема 4.2 Работа с фреймворками для разработки беспилотных летательных аппаратов на Python	4	4		2	2		
4.3	Тема 4.3 Создание беспилотного летательного аппарата на Python: от идеи до конечного продукта	5	4		1	3	1	
4.4	Тема 4.4 Разработка системы управления полетом беспилотного летательного аппарата на Python	4	3		1	2	1	
4.5	Тема 4.5 Разработка системы навигации для беспилотного летательного аппарата на Python	5	4		1	3	1	
4.6	Тема 4.6 Создание пользовательского интерфейса для управления беспилотным летательным аппаратом на Python	6	5		2	3	1	

4.7	Тема 4.7 Продвинутое программирование беспилотных летательных аппаратов на Python: машинное обучение, нейронные сети, искусственный интеллект	5	4	1	1	3	1	<b>Зачет</b>
5	Консультация	2	2			2		
	<b>Итого</b>	<b>144</b>	<b>120</b>	4	37	83	24	

## **II. ОЦЕНКА КАЧЕСТВА ОСВОЕНИЯ ПРОГРАММЫ (формы аттестации, оценочные и методические материалы)**

### **Условия прохождения курса**

**В курсе будут оцениваться:** текущая работа, включающая выполнение 4 практических задания. Задания будут оцениваться по пятибалльной шкале :

#### **Шкала оценивания:**

Нижнее значение - оценка «2».

Верхнее значение - оценка «5».

Минимальный проходной балл для успешной сдачи - оценка «3».

При оценке надо руководствоваться следующими критериями:

- 1) полнота и правильность выполнения заданий;
- 2) степень осознанности, понимания изученного;
- 3) оформление ответа.

#### **Оценка «5» ставится, если:**

- 1) Практическое задание выполнено без ошибок, полно изложен изученный материал;
- 2) учащийся понимает материал, может обосновать свои суждения и действия, хорошо применяет знания на практике, приводит необходимые примеры не только по учебному материалу, но и самостоятельно составленные;

**Оценка «4» ставится, если** ученик выполнил практическое задание, удовлетворяющее тем же требованиям, что и для оценки «5», но допускает 1—2 ошибки, которые сам же исправляет, и 1—2 недочета в последовательности излагаемого материала.

#### **Оценка «3» ставится, если** ученик освоил знания и понимает основы практического задания, но:

- 1) подготовил материал неполно и допускает неточности в последовательности выполнения заданий;
- 2) не может обосновать свои суждения и привести свои примеры;
- 3) допускает ошибки в оформлении.

**Оценка «2» ставится, если** ученик обнаруживает незнание большей части изучаемого материала, допускает ошибки в написании кода, беспорядочно и неуверенно излагает материал. Оценка «2» отмечает такие недостатки в подготовке ученика, которые являются серьезным препятствием к успешному овладению последующим материалом.

**Для успешного завершения обучения по курсу необходимо, чтобы 50 % заданий в рамках текущей работы были зачтены (оценены преподавателем на 3, 4 или 5 баллов).**

В таблице представлен перечень заданий, которые будут проводиться на практике в режиме онлайн с преподавателем в период обучения на курсе.

№ п/п	Мероприятия промежуточного контроля	Количество баллов
1.	Практическое задание № 1. Продвинутое программирование и управление беспилотными летательными аппаратами с использованием Python	2- 5
2.	Практическое задание № 2. Продвинутые техники программирования беспилотных летательных аппаратов на языке Python	2- 5
3.	Практическое задание № 3 Расширенные технологии и тенденции программирования беспилотных летательных аппаратов на языке Python	2- 5
4.	Практическое задание № 4. Практические примеры разработки беспилотных летательных аппаратов на языке Python	2- 5

В таблице представлен перечень заданий, которые необходимо выполнить в период обучения на курсе.

Формы проведения аттестации: зачетная работа.

### **Оценочные материалы для текущего контроля успеваемости:**

**Модуль 1.** Практическое задание № 1. Продвинутое программирование и управление беспилотными летательными аппаратами с использованием Python

Шаг 1. Определить координаты начальной точки, координаты конечной точки и скорость полета беспилотного летательного аппарата.

Шаг 2. Написать функцию, которая будет рассчитывать расстояние между двумя точками на плоскости по их координатам.

Шаг 3. Написать функцию, которая будет рассчитывать время полета беспилотного летательного аппарата до конечной точки по расстоянию и скорости полета.

Шаг 4. Написать функцию, которая будет рассчитывать направление полета беспилотного летательного аппарата по координатам начальной и конечной точек.

Шаг 5. Написать функцию, которая будет управлять движением беспилотного летательного аппарата по прямой до достижения конечной точки.

Шаг 6. Написать цикл, который будет вызывать функцию управления движением беспилотного летательного аппарата до тех пор, пока он не достигнет конечной точки.

Шаг 7. Вывести сообщение об успешном завершении полета.

Шаг 8. Проверить программу на работоспособность.

Шаг 9. Протестировать программу на нескольких наборах начальных и конечных координат.

Шаг 10. Оптимизировать программу, улучшив ее скорость и точность.

Шаг 11. Загрузите файл в Odin

## **Модуль 2.** Практическое задание № 2. Продвинутые техники программирования беспилотных летательных аппаратов на языке Python

Шаг 1. Создайте новый проект в вашей среде разработки, назовите его «Детектор объектов».

Шаг 2. Загрузите изображение с объектом, который вы хотите обнаружить, например, изображение машины.

Шаг 3. Используя библиотеку OpenCV для Python, напишите код, который будет обнаруживать машину на изображении. Воспользуйтесь, например, алгоритмом каскадных классификаторов Хаара (Haar Cascade).

Шаг 4. Добавьте код для вывода прямоугольной рамки вокруг найденного объекта (машины).

Шаг 5. Попробуйте запустить ваш код на других изображениях с машинами и проверьте, работает ли ваш детектор объектов правильно.

Шаг 6: Загрузите файл в Odin

## **Модуль 3.** Практическое задание № 3 Расширенные технологии и тенденции программирования беспилотных летательных аппаратов на языке Python

Шаг 1. Создайте класс для беспилотного летательного аппарата, который будет содержать основные характеристики, такие как высота, скорость, угол наклона, координаты, дистанция и т.д.

Шаг 2. Используйте библиотеку OpenCV для обработки изображений. Например, реализуйте распознавание объектов на изображениях, используя нейронные сети.

Шаг 3. Разработайте алгоритм управления беспилотным летательным аппаратом с помощью машинного обучения. Например, можно использовать алгоритмы глубокого обучения для обучения аппарата выполнять определенные действия в зависимости от условий и среды.

Шаг 4. Напишите код для управления беспилотным летательным аппаратом, используя библиотеку для управления полетом, такую как DroneKit. Шаг 5. Добавьте функциональность для получения и отправки данных, управления двигателем и т.д.

Шаг 6. Разработайте сценарий для тестирования беспилотного летательного аппарата. Например, можно создать сценарий для облета заданных объектов, съемки видео с высоты и т.д. Включите в сценарий проверку на ошибки и обработку исключительных ситуаций.

Шаг 7. Загрузите файл в Odin

## **Модуль 4.** Практическое задание № 4. Практические примеры разработки беспилотных летательных аппаратов на языке Python

Шаг 1. Создайте класс для беспилотного летательного аппарата, который будет содержать основные характеристики, такие как высота, скорость, угол наклона, координаты, дистанция и т.д.

Шаг 2. Напишите функцию для управления беспилотным летательным аппаратом. Функция должна использовать библиотеку для управления полетом, такую как Drone Kit, и добавлять функциональность для получения и отправки данных, управления двигателем и т.д.

Шаг 3. Разработайте сценарий для тестирования беспилотного летательного аппарата. Например, можно создать сценарий для облета заданных объектов, съемки видео с высоты и т.д. Включите в сценарий проверку на ошибки и обработку исключительных ситуаций.

Шаг 4. Используйте библиотеку OpenCV для обработки изображений. Например, реализуйте распознавание объектов на изображениях, используя нейронные сети.

Шаг 5. Разработайте алгоритм управления беспилотным летательным аппаратом с помощью машинного обучения. Например, можно использовать алгоритмы глубокого обучения для обучения аппарата выполнять определенные действия в зависимости от условий и среды

Шаг 6. Загрузите файл в Odin

**Для успешного завершения обучения по курсу необходимо, чтобы 50 % заданий в рамках текущей работы были зачтены (оценены преподавателем на 3, 4 или 5 баллов).**

### III. КАДРОВЫЕ УСЛОВИЯ

#### **Руководитель программы:**

**Сергеев Олег Владимирович,**

*Российский государственный социальный университет, Москва*

Факультет: Информационные технологии (бакалавриат)

Специальность: Педагогическое образование (информатика) (Сентябрь 2011 - Июль 2015)

*Российский государственный социальный университет, Москва*

Факультет: Информационные технологии (магистратура)

Специальность: Педагогическое образование (информатика) (Сентябрь 2016 - Июль 2019)

**Стаж работы** – более 7 лет.

#### **Сертификаты/ курсы:**

- Разработка веб- приложений с использованием Javascript ФГАОУ ВО «Национальный исследовательский университет «Высшая школа экономики». Московский институт электроники и математики им. А. Н. Тихонова 2019 год.
- Программирование веб- сайтов и веб- приложений на HTML и CSS ФГАОУ ВО «Национальный исследовательский университет «Высшая школа экономики». Московский институт электроники и математики им. А. Н. Тихонова 2018 год.
- Методика и использование 3d- прототипирования и моделирования на учебных занятиях в инженерном классе Академия «Просвещение» 2017 год.
- «Подготовка технических специалистов, оказывающих информационно- техническую помощь руководителю и организаторам пункта проведения экзамена при проведении государственной итоговой аттестации по образовательным программам основного общего образования» ГАОУ ДПО МЦКО 2017 год.
- «ИК технологии при обучении робототехнике» Городской методический центр Департамента образования города Москвы 2016 год  
Дизайнер- верстальщик РГСУ 2013 год
- «Методика обучения технологии программирования объектов дополненной реальности на Unity 3D». Городской Методический Центр 2023 год/

#### **IV. УСЛОВИЯ РЕАЛИЗАЦИИ ПРОГРАММЫ МОДУЛЯ (организационно- педагогические)**

##### **Материально- технические условия реализации программы:**

Обучение по программе реализовано в режиме синхронно- асинхронной работы слушателей в электронной среде. Для проведения синхронных занятий применяется программа видеоконференцсвязи Zoom. Дополнительно для организации работы слушателей могут использоваться Google- сервисы. Асинхронная работа слушателей реализуется на базе электронного курса на платформе электронного обучения LMS Odin.

##### **Требования к оборудованию:**

Сеть: скорость соединения от 2 Мб/с.

Оборудование для синхронных занятий: персональный компьютер (рекомендуется) / мобильный телефон / планшет; наушники, микрофон и камера (обязательно).

Для работы на платформе электронного обучения LMS Odin рекомендуется использовать персональный компьютер.

##### **Для успешной реализации Программы необходимо, чтобы рабочее место обучающегося и преподавателя включали в себя:**

- компьютеры, обеспечивающие возможность работы с мультимедийным контентом:
  - воспроизведение видеоизображений,
  - качественный стереозвук в наушниках,
  - речевой ввод с микрофона и др.
- периферийное оборудование:
  - принтер (черно/белой печати, формата А4);
  - устройства для ввода визуальной информации (сканер, цифровой фотоаппарат, web- камера и пр.);
  - устройства создания графической информации (графический планшет), использующиеся для создания и редактирования графических объектов, ввода рукописного текста;
  - акустические колонки;
  - оборудование, обеспечивающее подключение к сети Интернет (комплект оборудования для подключения к сети Интернет, сервер).
- программное обеспечение компьютера:
  - операционная система семейства MacOS или Windows;
  - программа IDLE, PyCharm
  - файловый менеджер (в составе операционной системы или др.);

- почтовый клиент (в составе операционных систем или др.);
- браузер (в составе операционных систем или др.);
- мультимедиа проигрыватель (в составе операционной системы или др.);
- антивирусная программа;
- программа- архиватор;
- программа- переводчик;
- программа интерактивного общения;
- текстовый редактор;
- растровый графический редактор;
- звуковой редактор;
- редактор Web- страниц.

### **Учебно- методическое и информационное обеспечение программы:**

#### **Методические рекомендации и пособия по изучению курса.**

Программа реализуется в формате смешанного обучения. Синхронные занятия включают интерактивные лекции и практические занятия, предусматривающие групповую и индивидуальную работу слушателей. Для организации асинхронной работы слушателей используются записи синхронных занятий, презентации, конспекты лекций, практические задания и тестовые вопросы, размещаемые в электронном курсе на платформе электронного обучения LMS Odin.

#### **Содержание комплекта учебно- методических материалов.**

Для изучения программы используется электронный курс в LMS Odin. Электронный курс включает информационно- организационные материалы по программе (учебно- тематический план, календарный график работы по программе, информацию о преподавателях программы, чат для объявлений и вопросов, сведения о результатах обучения), набор видеолекций, презентации к лекциям, набор ссылок на внешние ресурсы, систему заданий с инструкциями, списки основной и дополнительной литературы.

#### **Список литературы**

1. «Python для сложных задач: наука о данных и искусственный интеллект», Марко Гуззи, Эрик Шлутер, Инге Соргер, Джош Мингес, Дэн Велч, Энтони Де Барто, 2019 г.
2. «Python и машинное обучение: современное зрение компьютеров», Себастьян Рашка, 2017 г.
3. «Глубокое обучение на Python. Кeras, TensorFlow, PyTorch», Франсуа Шолле, 2019 г.
4. «Учим Python, делая крутые игры», Эрик Мэттиз, 2019 г.

5. «Python. Карманный справочник», Марк Лутц, 2018 г.

### *Дополнительный материал*

1. «Tkinter. Графический интерфейс на Python» (Карпов Б.). Книга «Tkinter. Графический интерфейс на Python» является хорошим введением в разработку на Tkinter. Она предоставляет понятное объяснение основных концепций и примеры кода.
  2. «Tkinter. Курс программирования графического интерфейса на Python» (ZetCode). На веб-сайте ZetCode есть подробный курс по программированию графического интерфейса на Python с использованием Tkinter. Курс содержит объяснения, примеры кода и упражнения.
  3. «Уроки Python: Tkinter» (Pythoner). Pythoner предлагает набор уроков по Tkinter, включая основы создания окон, работы с виджетами, обработку событий и многое другое. Уроки сопровождаются примерами кода и пояснениями.
  4. «Графический интерфейс на Python: Tkinter» (Иван Порубов). В блоге Ивана Порубова вы найдете серию статей о разработке графического интерфейса на Python с использованием Tkinter. Статьи охватывают различные аспекты Tkinter и содержат примеры кода.
  5. «Tkinter (Python) - разработка графического интерфейса» (ITVDN). На платформе ITVDN есть видеокурс по разработке графического интерфейса на Python с использованием Tkinter. Курс включает в себя теоретические объяснения и практические задания.
- Надеюсь, эти ресурсы помогут вам изучить Tkinter и углубиться в разработку графических интерфейсов с использованием этой библиотеки. Успехов в вашем проекте!

## Компетенции

Наименование компетенции	Тип компетенции	Знания, соответствующие компетенции	Умения, соответствующие компетенции	Владение инструментами, соответствующие компетенции
строка, не менее 10 символов	строка, значение из «общекультурные», «общепрофессиональные», «профессиональные»	строка не менее 50 символов, перечень знаний	строка не менее 50 символов, перечень умений	строка не менее 50 символов, перечень инструментов
ПК- 1 Создание программ на языке Python	общепрофессиональные	<p>Принципы работы информационных технологий.</p> <p>Принципы работы алгоритмов.</p> <p>Существующие структуры данных</p> <p>Принципы написания программ.</p> <p>Работа с командной строчкой.</p> <p>Подключение библиотек.</p>	<p>- Знание основ языка Python: понимание синтаксиса языка, структуры программ, работа с переменными, операторами, функциями и модулями.</p> <p>- Работа с датчиками и актуаторами БПЛА: понимание принципов работы датчиков и актуаторов, возможности и ограничения, умение настраивать и управлять ими через программу.</p> <p>- Работа с картами и навигацией: знание основ геопространственных систем, умение работать с картами и геоданными, настраивать и использовать системы навигации для БПЛА.</p> <p>- Разработка автономных полетов: понимание алгоритмов</p>	<p>- Python: уверенное владение языком программирования - Python является необходимым условием для программирования БПЛА. При работе с БПЛА также может быть полезным знание языков C++, MATLAB и других.</p> <p>- Интегрированная среда разработки (IDE): такие IDE, как PyCharm, Visual Studio Code, Sublime Text и - - Jupyter Notebook, могут значительно облегчить разработку программного обеспечения для БПЛА.</p> <p>- Библиотеки Python для работы с БПЛА: есть множество библиотек Python, которые могут</p>

			<p>планирования маршрута и управления полетом, умение создавать автономные программы для управления БПЛА.</p> <p>- Работа с аппаратным обеспечением: знание принципов работы датчиков, актуаторов, контроллеров и других устройств, умение подключать и настраивать их для работы с БПЛА.</p> <p>Для успешного овладения компетенцией необходимо иметь базовые знания по программированию и электронике, а также практические навыки работы с БПЛА и соответствующими инструментами и технологиями.</p>	<p>использоваться для работы с БПЛА, например, Dronekit, MAVLink, pymavlink и др.</p> <p>- Средства визуализации данных: Matplotlib, Seaborn, Plotly и другие библиотеки могут помочь в визуализации данных, полученных с БПЛА.</p> <p>- Средства обработки изображений: для обработки изображений, полученных с камер на БПЛА, можно использовать библиотеки Python, такие как OpenCV, Pillow и Scikit-image.</p>
--	--	--	---	--

### Модули

Наименование поля	Допустимые значения полей	Значение полей	Значение полей	Значение полей	Значение полей
Порядковый номер модуля	целое число	1	2	3	4
Название модуля	1.	Модуль 1: Продвинутое программирование и	Модуль 2: Продвинутые техники	Модуль 3: Расширенные технологии и тенденции	Модуль 4: Практические

		<b>управление беспилотными летательными аппаратами с использованием Python</b>	<b>программирования беспилотных летательных аппаратов на языке Python</b>	<b>программирования беспилотных летательных аппаратов на языке Python</b>	<b>примеры разработки беспилотных летательных аппаратов на языке Python</b>
<b>Описание модуля</b>	2.	<p>Модуль 1. В рамках прохождения данного модуля будут рассмотрены темы:</p> <p>Тема 1.1 Алгоритмы планирования траекторий для беспилотных летательных аппаратов.</p> <p>Тема 1.2 Реализация алгоритмов машинного обучения для автономного управления дронами</p> <p>Тема 1.3 Работа с компьютерным зрением и обработка изображений в контексте беспилотных летательных аппаратов</p> <p>Тема 1.4 Разработка системы оптической стабилизации камеры на дроне</p> <p>Тема 1.5 Использование датчиков и датчиков-обнаружителей (например, GPS, инфракрасные сенсоры, радары) для автономной навигации и избегания препятствий</p> <p>Тема 1.6 Реализация системы распознавания и классификации объектов с помощью машинного обучения на дроне</p>	<p>Модуль 2. В рамках прохождения данного модуля будут рассмотрены темы:</p> <p>Тема 2.1 Обзор продвинутых технологий беспилотных летательных аппаратов</p> <p>Тема 2.2 Работа с датчиками и сенсорами беспилотных летательных аппаратов</p> <p>Тема 2.3 Обработка видео и изображений с помощью Python для беспилотных летательных аппаратов</p> <p>Тема 2.4 Работа с базами данных для хранения данных беспилотных летательных аппаратов</p> <p>Тема 2.5 Разработка алгоритмов и программ для автоматического пилотирования беспилотных летательных аппаратов</p> <p>Тема 2.6 Использование машинного обучения и нейронных сетей в</p>	<p>Модуль 3. В рамках прохождения данного модуля будут рассмотрены темы:</p> <p>Тема 3.1 Разработка программного обеспечения для беспилотных летательных аппаратов на основе архитектур ROS и MAVLink</p> <p>Тема 3.2 Разработка и использование дополнительных средств связи между беспилотными летательными аппаратами и главным управляющим центром</p> <p>Тема 3.3 Разработка и применение технологий машинного обучения и искусственного интеллекта для беспилотных летательных аппаратов</p> <p>Тема 3.4 Интеграция различных датчиков и устройств для расширения функциональности беспилотных летательных аппаратов</p>	<p>Модуль 4. В рамках прохождения данного модуля будут рассмотрены темы:</p> <p>Тема 4.1 Основы разработки беспилотных летательных аппаратов на языке Python</p> <p>Тема 4.2 Работа с фреймворками для разработки беспилотных летательных аппаратов на Python</p> <p>Тема 4.3 Создание беспилотного летательного аппарата на Python: от идеи до конечного продукта</p> <p>Тема 4.4 Разработка системы управления полетом беспилотного летательного аппарата на Python</p> <p>Тема 4.5 Разработка системы навигации для беспилотного летательного аппарата на Python</p>

		<p>Тема 1.7 Оптимизация производительности беспилотных летательных аппаратов с использованием асинхронного программирования и многопоточности</p> <p>Тема 1.8 Разработка системы планирования миссий для беспилотных летательных аппаратов с учетом ограничений и целей</p> <p>Тема 1.9 Интеграция беспилотных летательных аппаратов с облачными сервисами для передачи данных, хранения информации и управления дронами из удаленного местоположения</p>	<p>беспилотных летательных аппаратах</p> <p>Тема 2.7 Разработка и настройка систем управления беспилотными летательными аппаратами</p> <p>Тема 2.8 Создание пользовательских интерфейсов для управления беспилотными летательными аппаратами</p> <p>Тема 2.9 Проект: Разработка программы для автоматического пилотирования беспилотного летательного аппарата на Python</p> <p>Тема 2.10 Проект: Разработка системы управления беспилотным летательным аппаратом на Python</p>	<p>Тема 3.5 Применение компьютерного зрения и обработки изображений для улучшения функциональности беспилотных летательных аппаратов</p> <p>Тема 3.6 Разработка и применение системы навигации и автоматического пилотирования на основе датчиков GPS и инерциальной навигации</p> <p>Тема 3.7 Разработка системы обнаружения и предотвращения столкновений для беспилотных летательных аппаратов</p> <p>Тема 3.8 Проект: Разработка приложения для управления беспилотным летательным аппаратом на основе архитектуры ROS и MAVLink</p> <p>Тема 3.9 Проект: Разработка системы обнаружения и предотвращения столкновений для беспилотных летательных аппаратов</p>	<p>Тема 4.6 Создание пользовательского интерфейса для управления беспилотным летательным аппаратом на Python</p> <p>Тема 4.7 Продвинутое техники разработки беспилотных летательных аппаратов на Python: машинное обучение, нейронные сети, искусственный интеллект</p>
<p><b>Аттестация по итогам модуля.</b> <b>Количество ак. часов</b></p>	<p><b>целое число</b></p>	<p><b>1</b></p>	<p><b>1</b></p>	<p><b>1</b></p>	<p><b>1</b></p>

<b>Аттестация по итогам модуля. Формы контроля</b>	<b>строка не менее 4 символов</b>	<b>Зачет</b>	<b>Зачет</b>	<b>Зачет</b>	<b>Зачет</b>
<b>Аттестация по итогам модуля. Диагностические инструменты</b>	<b>строка не менее 10 символов</b>	Практическое задание № 1. Модуль 1: Основы программирования беспилотных летательных аппаратов на языке Python	Практическое задание № 2. Модуль 2: Продвинутое программирование беспилотных летательных аппаратов на языке Python	Практическое задание № 3. Модуль 3: Расширенные технологии и тенденции программирования беспилотных летательных аппаратов на языке Python	Практическое задание № 4. Модуль 4: Практические примеры разработки беспилотных летательных аппаратов на языке Python
<b>Аттестация по итогам модуля. Показатели и критерии оценивания</b>	<b>строка не менее 50 символов</b>	<p>В ходе реализации программы предусмотрены следующие виды оценочных мероприятий:</p> <ul style="list-style-type: none"> <li>- практические задания и упражнения в период синхронной работы;</li> <li>- индивидуальные задания, в том числе со взаимной проверкой (комментированием, рецензированием).</li> </ul>			
<b>Аттестация по итогам модуля. Шкала оценивания, нижнее значение</b>	<b>целое число</b>	<b>2- 5</b>	<b>2- 5</b>	<b>2- 5</b>	<b>2- 5</b>
<b>Аттестация по итогам модуля. Шкала оценивания, верхнее значение</b>	<b>целое число</b>	<b>2- 5</b>	<b>2- 5</b>	<b>2- 5</b>	<b>2- 5</b>
<b>Аттестация по итогам модуля. Шкала оценивания, минимальный проходной балл для успешной сдачи</b>	<b>целое число в диапазоне шкалы</b>	<b>3</b>	<b>3</b>	<b>3</b>	<b>3</b>

## **РАБОЧАЯ ПРОГРАММА**

### **«Программирование БПЛА на Python: от теории к практике»**

#### **АННОТАЦИЯ**

«Программирование БПЛА на Python: от теории к практике» предназначена для тех, кто хочет научиться программированию беспилотных летательных аппаратов на языке Python. Мы начнем с теории управления беспилотными летательными аппаратами и основ программирования на Python, а затем перейдем к созданию реальных приложений для управления беспилотными летательными аппаратами. В программу входят лекции, демонстрационные видео, практические задания и проекты, которые помогут вам получить практические навыки в программировании беспилотных летательных аппаратов на Python. Вы изучите основные концепции и методы управления беспилотными летательными аппаратами, а также научитесь использовать библиотеки Python для работы с датчиками, выполнения задач автопилотирования и управления моторами и другими устройствами. После окончания программы вы будете иметь все необходимые знания и навыки для создания собственных управляющих ПО для беспилотных летательных аппаратов на языке Python. Программа направлена на формирование компетенций в соответствии с трудовыми функциями программиста.

В результате обучения выпускник программы будут иметь следующие компетенции:

- знание основ управления беспилотными летательными аппаратами, включая управление полетом, навигацию, автопилотирование и обнаружение препятствий;
- знание языка программирования Python и его применения в управлении беспилотными летательными аппаратами;
- умение использовать библиотеки Python для работы с датчиками, моторами и другими устройствами, необходимыми для управления беспилотными летательными аппаратами;
- умение разрабатывать управляющее ПО для беспилотных летательных аппаратов с использованием языка программирования Python;
- умение анализировать результаты полетов и улучшать управляющее ПО для повышения качества полетов и безопасности.

Модуль (описание)	Тема	Содержание	Вид учебных занятий	Объем в ак.ч.
<b>Модуль 1: Продвинутая разработка программного обеспечения для беспилотных летательных аппаратов</b>  Модуль 1: Продвинутая разработка программного обеспечения для беспилотных летательных аппаратов - это модуль обучения, направленный на развитие навыков разработки программного обеспечения для беспилотных летательных аппаратов (БПЛА) на более продвинутом уровне. В рамках этого модуля обучающиеся углубят свои знания в программировании на языке, таком как Python, и изучат продвинутые концепции и техники разработки ПО для БПЛА.	Тема 1.1 Алгоритмы планирования траекторий для беспилотных летательных аппаратов.(1)	Планирования траекторий для беспилотных летательных аппаратов.	теоретические занятия	(1 ч.)
	Тема 1.2 Реализация алгоритмов машинного обучения для автономного управления дронами(1)	Машинное обучение. автономное управление дронами. Алгоритмы	теоретические занятия	(1 ч.)
	Тема 1.3 Работа с компьютерным зрением и обработка изображений в контексте беспилотных летательных аппаратов (1)	Обработка снимков с бпла с помощью проективных алгоритмов. Обработка изображений в контексте беспилотных летательных аппаратов.	теоретические занятия	(1 ч.)
	Тема 1.4 Разработка системы оптической стабилизации камеры на дроне (7)	Обзор системы стабилизации камеры беспилотного летательного аппарата	теоретические занятия	(1 ч.)
		Задание к теме 1.4 Разработка системы оптической стабилизации камеры на дроне	практические занятия	(5 ч.)
	<a href="https://habr.com/ru/articles/240221/">https://habr.com/ru/articles/240221/</a>	самостоятельная работа	(1 ч.)	
Тема 1.5 Использование датчиков и датчиков- обнаружителей (например, GPS, инфракрасные сенсоры, радары) для автономной навигации и избегания препятствий	Приборы для автономной системы навигации и ориентирования беспилотных летательных аппаратов. GPS, инфракрасные сенсоры, радары	теоретические занятия	(1 ч.)	

<p>Они освоят принципы архитектуры программного обеспечения, разработку модульных систем, управление данными, обработку изображений и видео, а также взаимодействие с другими системами и устройствами.</p> <p>Обучающиеся также будут знакомиться с различными инструментами и фреймворками, используемыми в разработке программного обеспечения для БПЛА. По окончании модуля обучающиеся будут обладать продвинутыми навыками разработки программного обеспечения для БПЛА и смогут создавать сложные и функциональные системы управления и мониторинга для беспилотных летательных</p>		Задание к теме 1.5 Использование датчиков и датчиков- обнаружителей (например, GPS, инфракрасные сенсоры, радары) для автономной навигации и избегания препятствий	практические занятия	(5 ч.)	
		<u>Приборы для автономной системы навигации и ориентирования беспилотных летательных аппаратов   Статья в журнале «Молодой ученый» (moluch.ru)</u>	самостоятельная работа	(1 ч.)	
	<p>Тема 1.6 Реализация системы распознавания и классификации объектов с помощью машинного обучения на дроне (4)</p>		Модель быстрого и точного обнаружения объектов: обзор архитектуры SSD. Слои с многомасштабными функциями. Улучшения и недостатки SSD. Слежение за объектами	теоретические занятия	(1 ч.)
			Задание к теме 1.6 Реализация системы распознавания и классификации объектов с помощью машинного обучения на дроне	практические занятия	(2 ч.)
			<a href="https://habr.com/ru/articles/709432/">https://habr.com/ru/articles/709432/</a>	самостоятельная работа	(1 ч.)
	Тема 1.7 Оптимизация производительности беспилотных летательных аппаратов с использованием асинхронного программирования и многопоточности (5)		Многодисциплинарная оптимизация конструктивных параметров и динамики полетов БПЛА. Проектирование и оптимизация несущей системы квадрокоптера.	теоретические занятия	(1 ч.)

аппаратов.  (36 ч)		Задание к теме 1.7 Оптимизация производительности беспилотных летательных аппаратов с использованием асинхронного программирования и многопоточности	практические занятия	(3 ч.)
		<a href="https://cyberleninka.ru/article/n/optimizatsiya-funktsionirovaniya-bespilotnyh-letatelnyh-apparatov-snabzhennyh-solnechnymi-panelyami-v-rezhimah-podema-i-spuska">https://cyberleninka.ru/article/n/optimizatsiya-funktsionirovaniya-bespilotnyh-letatelnyh-apparatov-snabzhennyh-solnechnymi-panelyami-v-rezhimah-podema-i-spuska</a>	самостоятельная работа	(1 ч.)
	Тема 1.8 Разработка системы планирования миссий для беспилотных летательных аппаратов с учетом ограничений и целей (5)	Методы и алгоритмы управления. Оптимизация алгоритмов предварительного планирования применения беспилотного летательного аппарата.	теоретические занятия	(1 ч.)
		Задание к теме 1.8 Разработка системы планирования миссий для беспилотных летательных аппаратов с учетом ограничений и целей	практические занятия	(3 ч.)
		<a href="http://www.kg.ru/wp-content/uploads/2016/03/%D0%A1%D0%BA%D0%BE%D0%B1%D0%B5%D0%BB%D0%B5%D0%B2_%D0%AD%D0%A1%D0%98%D0%91_21-22_03_2017_Published.pdf">http://www.kg.ru/wp-content/uploads/2016/03/%D0%A1%D0%BA%D0%BE%D0%B1%D0%B5%D0%BB%D0%B5%D0%B2_%D0%AD%D0%A1%D0%98%D0%91_21-22_03_2017_Published.pdf</a>	самостоятельная работа	(1 ч.)
	1.9 Интеграция беспилотных летательных аппаратов с облачными сервисами для передачи данных,	Интеграция беспилотных летательных аппаратов с облачными сервисами для передачи данных, хранения информации и	теоретические занятия	(1)

	хранения информации и управления дронами из удаленного местоположения (5)	управления дронами из удаленного местоположения		
		<a href="https://core.ac.uk/download/pdf/38545214.pdf">https://core.ac.uk/download/pdf/38545214.pdf</a>	самостоятельная работа	(1 ч.)
	<b>Промежуточная аттестация</b>	<p><b>Промежуточная аттестация</b></p> <p><b>Практическое задание № 1. Основы программирования беспилотных летательных аппаратов на языке Python</b></p> <p>Шаг 1. Определить координаты начальной точки, координаты конечной точки и скорость полета беспилотного летательного аппарата.</p> <p>Шаг 2. Написать функцию, которая будет рассчитывать расстояние между двумя точками на плоскости по их координатам.</p> <p>Шаг 3. Написать функцию, которая будет рассчитывать время полета беспилотного летательного аппарата до конечной точки по расстоянию и скорости полета.</p> <p>Шаг 4. Написать функцию, которая будет рассчитывать направление полета беспилотного летательного аппарата по координатам начальной и конечной точек.</p> <p>Шаг 5. Написать функцию, которая будет управлять движением беспилотного летательного аппарата по прямой до достижения конечной точки.</p> <p>Шаг 6. Написать цикл, который будет вызывать функцию управления движением</p>	практические занятия	(3 ч.)

		беспилотного летательного аппарата до тех пор, пока он не достигнет конечной точки. Шаг 7. Вывести сообщение об успешном завершении полета. Шаг 8. Проверить программу на работоспособность. Шаг 9. Протестировать программу на нескольких наборах начальных и конечных координат. Шаг 10. Оптимизировать программу, улучшив ее скорость и точность. Шаг 11. Загрузите файл в Odin			
				<b>Объем в ак.ч.</b>	<b>Объем в %</b>
<b>ИТОГО:</b>			теоретические занятия	9	
			практические занятия	20	55%
			самостоятельная работа	6	16%
			аттестация	1	
			<b>Всего:</b>	<b>36</b>	

Модуль (описание)	Тема	Содержание	Вид учебных занятий	Объем в ак.ч.
<b>Модуль 2: Продвинутые техники программирования беспилотных летательных аппаратов на языке Python</b>  Модуль 2: Продвинутые техники программирования беспилотных летательных аппаратов на языке Python - это модуль обучения, нацеленный на расширение навыков программирования беспилотных летательных аппаратов (БПЛА) с использованием языка Python. В	Тема 2.1 Обзор продвинутых технологий беспилотных летательных аппаратов(2)	Обзор продвинутых технологий беспилотных летательных аппаратов. Классификация дронов. Рынок беспилотных летательных аппаратов.	теоретическое занятия	(1 ч.)
		<i>Отработка на практике полученных знаний. Задание к теме 2.1</i> Обзор продвинутых технологий беспилотных летательных аппаратов	практические занятия	(1 ч.)
	Тема 2.2 Работа с датчиками и сенсорами беспилотных летательных аппаратов (3)	Работа с датчиками и сенсорами беспилотных летательных аппаратов. Качество аэрофотосъемки. Точность. Подробный и детализированный сбор данных.	теоретическое занятия	(1 ч.)
		<i>Отработка на практике полученных знаний. Задание к теме 2.2</i> Работа с датчиками и сенсорами беспилотных летательных аппаратов	практические занятия	(2 ч.)
	Тема 2.3 Обработка видео и изображений с помощью Python для беспилотных летательных аппаратов (4)	Обработка видео и изображений с помощью Python для беспилотных летательных аппаратов. Обзор программных комплексов по обработке и анализу данных, полученных с помощью БПЛА.	теоретическое занятия	(1 ч.)
		<i>Отработка на практике полученных знаний. Задание к теме 2.3</i> 3 Обработка видео и изображений с помощью Python для беспилотных летательных аппаратов	практические занятия	(2 ч.)
		Изучить дополнительный материал по теме 2.3 3 Обработка видео и изображений с помощью Python для беспилотных летательных аппаратов	самостоятельная работа	(1 ч.)

<p>рамках этого модуля обучающиеся углубят свои знания в языке Python, изучат продвинутые концепции и техники программирования, применимые к разработке ПО для БПЛА. Они изучат объектно-ориентированное программирование (ООП), алгоритмы и структуры данных, многопоточность, работу с файлами и сетью, обработку и анализ данных и другие продвинутые темы. Обучающиеся также будут практиковаться в разработке сложных систем управления, обработки сенсорных данных, интеграции с внешними модулями и создании</p>		<a href="https://foresight.kubsau.ru/upload/iblock/858/85895bc9952dd9e5b58ce3ef20f12dac.pdf">https://foresight.kubsau.ru/upload/iblock/858/85895bc9952dd9e5b58ce3ef20f12dac.pdf</a>		
	<p>Тема 2.4 Работа с базами данных для хранения данных беспилотных летательных аппаратов (4)</p>	<p>Работа с базами данных для хранения данных беспилотных летательных аппаратов. Анализ БПЛА. Информационная система БПЛА. Android- приложение для БПЛА.</p>	теоретическое занятия	(1 ч.)
		<p><i>Отработка на практике полученных знаний.</i> <i>Задание к теме 2.4</i> Работа с базами данных для хранения данных беспилотных летательных аппаратов</p>	практические занятия	(2 ч.)
		<p>Изучить дополнительный материал по теме 2.4 Работа с базами данных для хранения данных беспилотных летательных аппаратов <a href="http://elibrary.asu.ru/xmlui/bitstream/handle/asu/9542/vkr.pdf?sequence=1&amp;isAllowed=y">http://elibrary.asu.ru/xmlui/bitstream/handle/asu/9542/vkr.pdf?sequence=1&amp;isAllowed=y</a></p>	самостоятельная работа	(1 ч.)
	<p>Тема 2.5 Разработка алгоритмов и программ для автоматического пилотирования беспилотных летательных аппаратов (4)</p>	<p>Разработка алгоритмов и программ для автоматического пилотирования беспилотных летательных аппаратов. Моделирование задач управления полетом БПЛА с высоким углом атаки. Алгоритмы работы многофункционального комплекса управления БПЛА.</p>	теоретическое занятия	(1 ч.)
		<p><i>Отработка на практике полученных знаний.</i> <i>Задание к теме 2.5</i> Разработка алгоритмов и программ для автоматического пилотирования беспилотных летательных аппаратов</p>	практические занятия	(2 ч.)

<p>устойчивых и надежных программных решений для БПЛА. По окончании модуля обучающиеся будут обладать продвинутыми навыками программирования на языке Python для разработки сложных и эффективных систем для управления беспилотными летательными аппаратами.</p> <p><b>(36 ч)</b></p>		<p>Изучить дополнительный материал по теме 2.5 Разработка алгоритмов и программ для автоматического пилотирования беспилотных летательных аппаратов</p> <p><a href="https://tekhnosfera.com/issledovanie-i-razrabotka-metodov-sistem-i-algoritmov-avtomaticheskogo-upravleniya-bespilotnymi-sredstvami-monitoringa">https://tekhnosfera.com/issledovanie-i-razrabotka-metodov-sistem-i-algoritmov-avtomaticheskogo-upravleniya-bespilotnymi-sredstvami-monitoringa</a></p>	самостоятельная работа	(1 ч.)
	<p>Тема 2.6 Использование машинного обучения и нейронных сетей в беспилотных летательных аппаратах (4)</p>	<p>Использование машинного обучения и нейронных сетей в беспилотных летательных аппаратах. Алгоритм обучения нейронной сети. Параметры навигационных систем. Методы навигации.</p>	теоретические занятия	(1 ч.)
		<p><i>Отработка на практике полученных знаний.</i> <i>Задание к теме 2.6</i></p> <p>Использование машинного обучения и нейронных сетей в беспилотных летательных аппаратах</p>	практические занятия	(3 ч.)
	<p>Тема 2.7 Разработка и настройка систем управления беспилотными летательными аппаратами (3)</p>	<p>Разработка и настройка систем управления беспилотными летательными аппаратами. Разработка бортового общесистемного и функционального программного обеспечения. Алгоритмы работы с системами астронавигации и ориентации.</p>	теоретические занятия	(1 ч.)
		<p><i>Отработка на практике полученных знаний.</i> <i>Задание к теме 2.7</i></p> <p>Разработка и настройка систем управления беспилотными летательными аппаратами</p>	практические занятия	(2 ч.)
	<p>Тема 2.8 Создание</p>	<p>Создание пользовательских интерфейсов для управления беспилотными летательными аппаратами. Отрасли, в которых</p>	теоретические занятия	(1 ч.)

	пользовательских интерфейсов для управления беспилотными летательными аппаратами (4)	используются дроны. Архитектура программного обеспечения для БПЛА.		
		<i>Отработка на практике полученных знаний.</i> <i>Задание к теме 2.8</i> Создание пользовательских интерфейсов для управления беспилотными летательными аппаратами	практические занятия	(2 ч.)
		Изучить дополнительный материал по теме 2.8 Создание пользовательских интерфейсов для управления беспилотными летательными аппаратами <a href="https://goodbit.dev/blog/soveti-i-prognozi-po-razrabotke-programmnogo-obespecheniya-dlya-bespilotnih-letatelnyh-apparatov">https://goodbit.dev/blog/soveti-i-prognozi-po-razrabotke-programmnogo-obespecheniya-dlya-bespilotnih-letatelnyh-apparatov</a>	самостоятельная работа	(1 ч.)
	Тема 2.9 Проект: Разработка программы для автоматического пилотирования беспилотного летательного аппарата на Python (4)	Проект: Разработка программы для автоматического пилотирования беспилотного летательного аппарата на Python. Программирование полёта. Компьютерное зрение.	теоретические занятия	(1 ч.)
		<i>Отработка на практике полученных знаний.</i> <i>Задание к теме 2.9</i> Проект: Разработка программы для автоматического пилотирования беспилотного летательного аппарата на Python	практические занятия	(2 ч.)
		Изучить дополнительный материал по теме 2.9 Проект: Разработка программы для автоматического пилотирования беспилотного летательного аппарата на Python <a href="https://www.youtube.com/watch?v=Ng6cLh-Hyvg">https://www.youtube.com/watch?v=Ng6cLh-Hyvg</a>	самостоятельная работа	(1 ч.)
	Тема 2.10 Проект: Разработка	Изучить дополнительный материал по теме 2.10 Проект: Разработка системы управления беспилотным летательным аппаратом на Python	самостоятельная работа	(1 ч.)

	системы управления беспилотным летательным аппаратом на Python (4)	<a href="https://www.mathnet.ru/links/986a0e1dcc3f8c4d55687c81149e4b3b/itvs127.pdf">https://www.mathnet.ru/links/986a0e1dcc3f8c4d55687c81149e4b3b/itvs127.pdf</a>			
	Промежуточная аттестация	<p><b>Практическое задание № 2.</b></p> <p>Продвинутые техники программирования беспилотных летательных аппаратов на языке Python</p> <p>Шаг 1. Создайте новый проект в вашей среде разработки, назовите его «Детектор объектов».</p> <p>Шаг 2. Загрузите изображение с объектом, который вы хотите обнаружить, например, изображение машины.</p> <p>Шаг 3. Используя библиотеку OpenCV для Python, напишите код, который будет обнаруживать машину на изображении. Воспользуйтесь, например, алгоритмом каскадных классификаторов Хаара (Haar Cascade).</p> <p>Шаг 4. Добавьте код для вывода прямоугольной рамки вокруг найденного объекта (машины).</p> <p>Шаг 5. Попробуйте запустить ваш код на других изображениях с машинами и проверьте, работает ли ваш детектор объектов правильно.</p> <p>Шаг 6: Загрузите файл в Odin</p>	практические занятия	(3 ч.)	
				<b>Объем в ак.ч.</b>	<b>Объем в %</b>
<b>ИТОГО:</b>			теоретические занятия	9	
			практические занятия	20	55%

	самостоятельная работа	6	17%
	аттестация	1	
	<b>Всего:</b>	<b>36</b>	

Модуль (описание)	Тема	Содержание	Вид учебных занятий	Объем в ак.ч.
<b>Модуль 3: Расширенные технологии и тенденции программирования беспилотных летательных аппаратов на языке Python</b>  Модуль 3: Расширенные технологии и тенденции программирования беспилотных летательных аппаратов на языке Python - это модуль	Тема 3.1 Разработка программного обеспечения для беспилотных летательных аппаратов на основе архитектур ROS и MAVLink	Разработка программного обеспечения для беспилотных летательных аппаратов на основе архитектур ROS и MAVLink. Полетный контроллер. Архитектура программного и аппаратного обеспечения.	теоретические занятия	(1 ч.)
		<i>Отработка на практике полученных знаний. Задание к теме 3.1</i> Разработка программного обеспечения для беспилотных летательных аппаратов на основе архитектур ROS и MAVLink	практические занятия	(2 ч.)
		Изучить дополнительный материал по теме 3.1 Разработка программного обеспечения для беспилотных летательных аппаратов на основе архитектур ROS и MAVLink <a href="https://cyberleninka.ru/article/n/razrabotka-arhitektury-poletnogo-kontrollera-dlya-kvadrokoptera-na-baze-odnoplavnoy-kompyutera-raspberry-pi/viewer">https://cyberleninka.ru/article/n/razrabotka-arhitektury-poletnogo-kontrollera-dlya-kvadrokoptera-na-baze-odnoplavnoy-kompyutera-raspberry-pi/viewer</a>	самостоятельная работа	(1 ч.)

<p>обучения, посвященный изучению передовых технологий и актуальных тенденций в программировании беспилотных летательных аппаратов (БПЛА) с использованием языка Python. В рамках этого модуля обучающиеся будут ознакомлены с новыми разработками и инновационными подходами к программированию БПЛА. Они изучат расширенные технологии, такие как машинное обучение, компьютерное зрение, нейронные сети и обработка естественного языка,</p>	<p>Тема 3.2 Разработка и использование дополнительных средств связи между беспилотными летательными аппаратами и главным управляющим центром</p>	<p>Разработка и использование дополнительных средств связи между беспилотными летательными аппаратами и главным управляющим центром. Применение перспективных навигационных технологий в пилотажно- навигационной аппаратуре.</p>	теоретические занятия	(1 ч.)
		<p><i>Отработка на практике полученных знаний.</i> <i>Задание к теме 3.2</i> Разработка и использование дополнительных средств связи между беспилотными летательными аппаратами и главным управляющим центром</p>	практические занятия	(1 ч.)
		<p>Изучить дополнительный материал по теме 3.2 Разработка и использование дополнительных средств связи между беспилотными летательными аппаратами и главным управляющим центром <a href="https://function.mil.ru/files/morf/Sbornik_dokladov_konferencii_bla.pdf">https://function.mil.ru/files/morf/Sbornik_dokladov_konferencii_bla.pdf</a></p>	самостоятельная работа	(1 ч.)
	<p>Тема 3.3 Разработка и применение технологий машинного обучения и искусственного интеллекта для беспилотных летательных аппаратов</p>	<p>Разработка и применение технологий машинного обучения и искусственного интеллекта для беспилотных летательных аппаратов. Изучение основ системы искусственного интеллекта насекомых дрона повышения производительности дронов.</p>	теоретические занятия	(1 ч.)
		<p><i>Отработка на практике полученных знаний.</i> <i>Задание к теме 3.3</i> Разработка и применение технологий машинного обучения и искусственного интеллекта для беспилотных летательных аппаратов</p>	практические занятия	(2 ч.)

<p>и узнают, как эти технологии применяются в разработке ПО для БПЛА. Обучающиеся также будут рассмотрены современные тенденции в области программирования БПЛА, такие как использование библиотек и фреймворков, разработка архитектуры масштабируемых систем и облачных решений. По окончании модуля обучающиеся будут иметь расширенные знания и навыки в программировании беспилотных летательных аппаратов на языке Python, а также будут знакомы с</p>		<p>Изучить дополнительный материал по теме 3.3 Разработка и применение технологий машинного обучения и искусственного интеллекта для беспилотных летательных аппаратов  <a href="https://ts2.space/ru/%D0%BA%D0%B0%D0%BA-%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%D0%B5%D1%82-%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0-%D0%B8%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%BE%D0%B3%D0%BE/">https://ts2.space/ru/%D0%BA%D0%B0%D0%BA-%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%D0%B5%D1%82-%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0-%D0%B8%D1%81%D0%BA%D1%83%D1%81%D1%81%D1%82%D0%B2%D0%B5%D0%BD%D0%BD%D0%BE%D0%B3%D0%BE/</a></p>	самостоятельная работа	(1 ч.)
	Тема 3.4 Интеграция различных датчиков и устройств для расширения функциональности беспилотных летательных аппаратов	<p>Интеграция различных датчиков и устройств для расширения функциональности беспилотных летательных аппаратов. Передовые датчики и исполнительные механизмы на основе биотехнологий. Роль биосенсоров и приводов в улучшении восприятия и управления дроном.</p>	теоретические занятия	(1 ч.)
		<p>Изучить дополнительный материал по теме 3.4 Интеграция различных датчиков и устройств для расширения функциональности беспилотных летательных аппаратов  <a href="https://ts2.space/ru/%D0%BA%D0%B0%D0%BA%D0%BE%D0%B2%D1%8B-%D0%BE%D0%B6%D0%B8%D0%B4%D0%B0%D0%B5%D0%BC%D1%8B%D0%B5-%D0%B2%D0%B0%D1%80%D0%B8%D0%B0%D0%BD%D1%82%D1%8B-">https://ts2.space/ru/%D0%BA%D0%B0%D0%BA%D0%BE%D0%B2%D1%8B-%D0%BE%D0%B6%D0%B8%D0%B4%D0%B0%D0%B5%D0%BC%D1%8B%D0%B5-%D0%B2%D0%B0%D1%80%D0%B8%D0%B0%D0%BD%D1%82%D1%8B-</a></p>	самостоятельная работа	(1 ч.)

<p>передовыми технологиями и тенденциями в этой области.</p> <p><b>(36 ч)</b></p>		<a href="#"><u>%D0%B8%D0%BD%D1%82%D0%B5%D0%B3%D1%80%D0%B0%D1%86%D0%B8- 33/</u></a>		
	<p>Тема 3.5 Применение компьютерного зрения и обработки изображений для улучшения функциональности беспилотных летательных аппаратов</p>	<p>Применение компьютерного зрения и обработки изображений для улучшения функциональности беспилотных летательных аппаратов. Взаимосвязь машинного зрения с компьютерным зрением.</p>	теоретические занятия	(1 ч.)
		<p><i>Отработка на практике полученных знаний.</i> <i>Задание к теме 3.5</i> Применение компьютерного зрения и обработки изображений для улучшения функциональности беспилотных летательных аппаратов</p>	практические занятия	(2 ч.)
	<p>Тема 3.6 Разработка и применение системы навигации и автоматического пилотирования на основе датчиков GPS и инерциальной навигации</p>	<p>Разработка и применение системы навигации и автоматического пилотирования на основе датчиков GPS и инерциальной навигации. Система автоматического управления.</p>	теоретические занятия	(1 ч.)
		<p><i>Отработка на практике полученных знаний.</i> <i>Задание к теме 3.6</i> Разработка и применение системы навигации и автоматического пилотирования на основе датчиков GPS и инерциальной навигации</p>	практические занятия	(3 ч.)
	<p>Тема 3.7 Разработка системы</p>	<p>Разработка системы обнаружения и предотвращения столкновений для беспилотных летательных аппаратов. Технологии предотвращения столкновений дронов. Интеллектуальные системы</p>	теоретические занятия	(1 ч.)

	обнаружения и предотвращения столкновений для беспилотных летательных аппаратов	предотвращения столкновений, основанные на лидарных и ToF- системах.		
		<i>Отработка на практике полученных знаний.</i> Задание к теме 3.7 Разработка системы обнаружения и предотвращения столкновений для беспилотных летательных аппаратов	практические занятия	(3 ч.)
	Тема 3.8 Проект: Разработка приложения для управления беспилотным летательным аппаратом на основе архитектуры ROS и MAVLink	Проект: Разработка приложения для управления беспилотным летательным аппаратом на основе архитектуры ROS и MAVLink. Алгоритм управления.	теоретические занятия	(1 ч.)
		<i>Отработка на практике полученных знаний.</i> Задание к Теме 3.8 Проект: Разработка приложения для управления беспилотным летательным аппаратом на основе архитектуры ROS и MAVLink	практические занятия	(2 ч.)
		Изучить дополнительный материал по теме 3.8 Проект: Разработка приложения для управления беспилотным летательным аппаратом на основе архитектуры ROS и MAVLink <a href="https://se.math.spbu.ru/thesis/texts/Ahremchik_Jan_Valer%27evich_Master_Thesis_2021_text.pdf">https://se.math.spbu.ru/thesis/texts/Ahremchik_Jan_Valer%27evich_Master_Thesis_2021_text.pdf</a>	самостоятельная работа	(1 ч.)
	Тема 3.9 Разработка системы обнаружения и предотвращения столкновений	Разработка системы обнаружения и предотвращения столкновений для беспилотных летательных аппаратов. Как работает система обнаружения и уклонения дрона.	теоретические занятия	(2 ч.)
		<i>Отработка на практике полученных знаний.</i>	практические занятия	(2 ч.)

	для беспилотных летательных аппаратов	<p><i>Задание к Теме 3.9</i></p> <p>Разработка системы обнаружения и предотвращения столкновений для беспилотных летательных аппаратов</p>		
		<p>Изучить дополнительный материал по теме 3.9</p> <p>Разработка системы обнаружения и предотвращения столкновений для беспилотных летательных аппаратов</p> <p><a href="https://ts2.space/ru/%D0%BA%D0%B0%D0%BA-%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%D0%B5%D1%82-%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0-%D0%BE%D0%B1%D0%BD%D0%B0%D1%80%D1%83%D0%B6%D0%B5%D0%BD%D0%B8%D1%8F-%D0%B8-%D1%83%D0%BA/">https://ts2.space/ru/%D0%BA%D0%B0%D0%BA-%D1%80%D0%B0%D0%B1%D0%BE%D1%82%D0%B0%D0%B5%D1%82-%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B0-%D0%BE%D0%B1%D0%BD%D0%B0%D1%80%D1%83%D0%B6%D0%B5%D0%BD%D0%B8%D1%8F-%D0%B8-%D1%83%D0%BA/</a></p>	самостоятельная работа	(1 ч.)
	Промежуточная аттестация	<p><b>Практическое задание № 3.</b> Расширенные технологии и тенденции программирования беспилотных летательных аппаратов на языке Python</p> <p>Шаг 1. Создайте класс для беспилотного летательного аппарата, который будет содержать основные характеристики, такие как высота, скорость, угол наклона, координаты, дистанция и т.д.</p> <p>Шаг 2. Используйте библиотеку OpenCV для обработки изображений. Например, реализуйте распознавание объектов на изображениях, используя нейронные сети.</p> <p>Шаг 3. Разработайте алгоритм управления беспилотным летательным аппаратом с помощью</p>	практические занятия	(3 ч.)

		<p>машинного обучения. Например, можно использовать алгоритмы глубокого обучения для обучения аппарата выполнять определенные действия в зависимости от условий и среды.</p> <p>Шаг 4. Напишите код для управления беспилотным летательным аппаратом, используя библиотеку для управления полетом, такую как DroneKit. Шаг 5. Добавьте функциональность для получения и отправки данных, управления двигателем и т.д.</p> <p>Шаг 6. Разработайте сценарий для тестирования беспилотного летательного аппарата. Например, можно создать сценарий для облета заданных объектов, съемки видео с высоты и т.д. Включите в сценарий проверку на ошибки и обработку исключительных ситуаций.</p> <p>Шаг 7. Загрузите файл в Odin</p>			
				<b>Объем в ак.ч.</b>	<b>Объем в %</b>
<b>ИТОГО:</b>			теоретические занятия	10	
			практические занятия	19	53%
			самостоятельная работа	6	17%
			аттестация	1	
			<b>Всего:</b>	<b>36</b>	

Модуль (описание)	Тема	Содержание	Вид учебных занятий	Объем в ак.ч.
-------------------	------	------------	---------------------	---------------

<p><b>Модуль 4: Практические примеры разработки беспилотных летательных аппаратов на языке Python</b></p> <p>Модуль 4: Практические примеры разработки беспилотных летательных аппаратов на языке Python - это модуль обучения, ориентированный на применение полученных знаний и навыков в практических задачах разработки беспилотных летательных аппаратов (БПЛА) с использованием языка Python. В рамках этого модуля</p>	<p>Тема 4.1 Основы разработки беспилотных летательных аппаратов на языке Python</p>	<p>Основы разработки беспилотных летательных аппаратов на языке Python. Обоснование выбора средств и методов разработки. Особенности реализации используемых алгоритмов обработки и вывода данных.</p>	теоретические занятия	(1 ч.)
		<p><i>Отработка на практике полученных знаний. Задание к теме 4.1</i> Основы разработки беспилотных летательных аппаратов на языке Python</p>	практические занятия	(3 ч.)
		<p>Изучить дополнительный материал по теме 4.1 Основы разработки беспилотных летательных аппаратов на языке Python <a href="https://nauchkor.ru/pubs/razrabotka-programmnogo-modulya-vizualizatsii-izobrazheniy-mestnosti-na-osnove-dannyh-bespilotnogo-letatel'nogo-apparata-5ab272087966e145c68c1134">https://nauchkor.ru/pubs/razrabotka-programmnogo-modulya-vizualizatsii-izobrazheniy-mestnosti-na-osnove-dannyh-bespilotnogo-letatel'nogo-apparata-5ab272087966e145c68c1134</a></p>	самостоятельная работа	(1 ч.)
	<p>Тема 4.2 Работа с фреймворками для разработки беспилотных летательных аппаратов на Python</p>	<p>Работа с фреймворками для разработки беспилотных летательных аппаратов на Python. Фреймворки и библиотеки Python. Возможности фреймворков.</p>	теоретические занятия	(2 ч.)
		<p><i>Отработка на практике полученных знаний. Задание к теме 4.2</i> Работа с фреймворками для разработки беспилотных летательных аппаратов на Python</p>	практические занятия	(2 ч.)
	<p>Тема 4.3 Создание беспилотного летательного аппарата</p>	<p>Создание беспилотного летательного аппарата на Python: от идеи до конечного продукта. Базовые элементы, системы управления, стабилизации, навигации и ориентации.</p>	теоретические занятия	(1 ч.)

<p>обучающиеся будут решать реальные задачи и выполнять проекты, связанные с разработкой ПО для БПЛА. Они будут работать над созданием полетных систем, автономных функций, обработкой сенсорных данных, управлением полетом и другими аспектами, необходимыми для успешной разработки беспилотных летательных аппаратов. Обучающиеся будут применять свои знания в языке Python, программировании и алгоритмах для разработки функциональных и эффективных систем управления БПЛА.</p>	<p>аппарата на Python: от идеи до конечного продукта</p>	<p><i>Отработка на практике полученных знаний.</i> <i>Задание к теме 4.3</i> Создание беспилотного летательного аппарата на Python: от идеи до конечного продукта</p>	<p>практические занятия</p>	<p>(3 ч.)</p>
		<p>Изучить дополнительный материал по теме 4.3 Создание беспилотного летательного аппарата на Python: от идеи до конечного продукта <a href="https://gosniias.ru/pages/d/st-4-konf.pdf">https://gosniias.ru/pages/d/st-4-konf.pdf</a></p>	<p>самостоятельная работа</p>	<p>(1 ч.)</p>
	<p>Тема 4.4 Разработка системы управления полетом беспилотного летательного аппарата на Python</p>	<p>Разработка системы управления полетом беспилотного летательного аппарата на Python. Автономная навигация, методы коррекции, алгоритмы комплексирования, картографирование.</p>	<p>теоретические занятия</p>	<p>(1 ч.)</p>
		<p><i>Отработка на практике полученных знаний.</i> <i>Задание к теме 4.4</i> Разработка системы управления полетом беспилотного летательного аппарата на Python</p>	<p>практические занятия</p>	<p>(2 ч.)</p>
		<p>Изучить дополнительный материал по теме 4.4 Разработка системы управления полетом беспилотного летательного аппарата на Python <a href="https://gosniias.ru/pages/d/st-4-konf.pdf">https://gosniias.ru/pages/d/st-4-konf.pdf</a></p>	<p>самостоятельная работа</p>	<p>(1 ч.)</p>
	<p>Тема 4.5 Разработка системы навигации для беспилотного летательного</p>	<p>Разработка системы навигации для беспилотного летательного аппарата на Python. Модель инерциальной навигации для беспилотных летательных аппаратов.</p>	<p>теоретические занятия</p>	<p>(1 ч.)</p>
		<p><i>Отработка на практике полученных знаний.</i> <i>Задание к теме 4.5</i></p>	<p>практические занятия</p>	<p>(3 ч.)</p>

<p>По окончании модуля обучающиеся будут иметь опыт работы с реальными примерами разработки беспилотных летательных аппаратов на языке Python и будут готовы к практическому применению своих навыков в сфере авиационных технологий.</p> <p><b>(36 ч)</b></p>	аппарата на Python	Разработка системы навигации для беспилотного летательного аппарата на Python		
		Изучить дополнительный материал по теме 4.5 Разработка системы навигации для беспилотного летательного аппарата на Python <a href="https://7universum.com/ru/tech/archive/item/11683">https://7universum.com/ru/tech/archive/item/11683</a>	самостоятельная работа	(1 ч.)
	Тема 4.6 Создание пользовательского интерфейса для управления беспилотным летательным аппаратом на Python	Создание пользовательского интерфейса для управления беспилотным летательным аппаратом на Python. Составление алгоритмов для решения прикладных задач. Реализация алгоритмов на компьютере в виде программ, написанных на языке Python.	теоретические занятия	(2 ч.)
		<i>Отработка на практике полученных знаний.</i> <i>Задание к теме 4.6</i> Создание пользовательского интерфейса для управления беспилотным летательным аппаратом на Python	практические занятия	(3 ч.)
		Изучить дополнительный материал по теме 4.6 Создание пользовательского интерфейса для управления беспилотным летательным аппаратом на Python <a href="https://nauchkor.ru/pubs/razrabotka-programmnogo-modulya-vizualizatsii-izobrazheniy-mestnosti-na-osnove-dannyh-bespilotnogo-letatel'nogo-apparata-5ab272087966e145c68c1134">https://nauchkor.ru/pubs/razrabotka-programmnogo-modulya-vizualizatsii-izobrazheniy-mestnosti-na-osnove-dannyh-bespilotnogo-letatel'nogo-apparata-5ab272087966e145c68c1134</a>	самостоятельная работа	(1 ч.)
Тема 4.7 Продвинутые техники разработки	Продвинутые техники разработки беспилотных летательных аппаратов на Python: машинное обучение, нейронные сети, искусственный интеллект. Информационно-управляющий комплекс бортового	теоретические занятия	(1 ч.)	

	беспилотных летательных аппаратов на Python: машинное обучение, нейронные сети, искусственный интеллект	оборудования беспилотных летательных аппаратов.  Изучить дополнительный материал по теме 4.7 Продвинутые техники разработки беспилотных летательных аппаратов на Python: машинное обучение, нейронные сети, искусственный интеллект. <a href="https://mil.ru/files/morf/Sbornik-konferencii-2017.pdf">https://mil.ru/files/morf/Sbornik-konferencii-2017.pdf</a>	самостоятельная работа	(1 ч.)
Консультация				(2 ч.)
	Промежуточная аттестация	<b>Практическое задание № 4.</b> Практические примеры разработки беспилотных летательных аппаратов на языке Python Шаг 1. Создайте класс для беспилотного летательного аппарата, который будет содержать основные характеристики, такие как высота, скорость, угол наклона, координаты, дистанция и т.д. Шаг 2. Напишите функцию для управления беспилотным летательным аппаратом. Функция должна использовать библиотеку для управления полетом, такую как Drone Kit, и добавлять функциональность для получения и отправки данных, управления двигателем и т.д. Шаг 3. Разработайте сценарий для тестирования беспилотного летательного аппарата. Например, можно создать сценарий для облета заданных объектов, съемки видео с высоты и т.д. Включите в сценарий проверку на ошибки и обработку исключительных ситуаций.	практические занятия	(3 ч.)

		<p>Шаг 4. Используйте библиотеку OpenCV для обработки изображений. Например, реализуйте распознавание объектов на изображениях, используя нейронные сети.</p> <p>Шаг 5. Разработайте алгоритм управления беспилотным летательным аппаратом с помощью машинного обучения. Например, можно использовать алгоритмы глубокого обучения для обучения аппарата выполнять определенные действия в зависимости от условий и среды</p> <p>Шаг 6. Загрузите файл в Odin</p>			
				<b>Объем в ак.ч.</b>	<b>Объем в %</b>
<b>ИТОГО:</b>			теоретические занятия	9	
			практические занятия	20	55%
			самостоятельная работа	6	17%
			аттестация	1	
			<b>Всего:</b>	<b>36</b>	

#### ***Задание к теме 1.4***

*Разработка системы оптической стабилизации камеры на дроне (5 ч)*

1. Изучение и анализ: Изучите основные алгоритмы и методы стабилизации камеры, применяемые в аэрокосмической и робототехнике. Ознакомьтесь с библиотеками компьютерного зрения, доступными на языке Python, такими как OpenCV, и изучите их возможности в контексте стабилизации камеры на дроне. Проанализируйте требования к системе стабилизации и определите наиболее подходящий алгоритм или комбинацию алгоритмов для данного проекта.
2. Разработка программы: На основе алгоритма стабилизации, выбранного на предыдущем этапе, разработайте программу на языке Python. Используйте библиотеку OpenCV для обработки видеопотока с камеры на дроне. Реализуйте алгоритм стабилизации, который будет анализировать каждый кадр видеопотока и корректировать позицию камеры для компенсации движений дрона. Учтите возможные шумы и помехи в видеопотоке и предусмотрите адаптивность системы к изменяющимся условиям полета.
3. Тестирование и отладка: Протестируйте разработанную программу с использованием тестовых видеозаписей или в реальных условиях полета дрона. Оцените эффективность системы стабилизации, проверьте, насколько успешно камера компенсирует движения дрона. Выявите возможные проблемы и ошибки в программе и проведите отладку, чтобы устранить их. Проведите несколько тестовых полетов, чтобы убедиться в надежности и стабильности работы системы.
4. Улучшение и оптимизация: Проведите анализ результатов тестирования и выявите области, где система стабилизации камеры может быть улучшена. Рассмотрите возможность оптимизации алгоритмов для более быстрой и точной стабилизации. Используйте профайлеры и инструменты для оптимизации производительности Python, чтобы улучшить скорость выполнения программы. Проведите повторные тесты и сравните результаты с предыдущими, чтобы оценить эффективность внесенных изменений.

#### ***Задание к теме 1.5***

*Использование датчиков и датчиков-обнаружителей (например, GPS, инфракрасные сенсоры, радары) для автономной навигации и избегания препятствий (5 ч.)*

1. Изучение документации и API: Найдите и изучите документацию открытого API для работы с датчиками и датчиками-обнаружителями, такими как GPS, инфракрасные сенсоры и радары. Опишите основные методы и параметры, которые доступны для использования через это API. Понимание функциональности и возможностей датчиков поможет вам в разработке программы для автономной навигации и избегания препятствий.
2. Разработка программы на Python: Напишите программу на языке Python, которая будет использовать выбранное API для работы с датчиками и датчиками-обнаружителями. Программа должна получать данные от датчиков и анализировать их для определения текущего положения и обнаружения препятствий. Реализуйте алгоритмы для автономной навигации и принятия решений об избегании препятствий на основе этих данных. Программа должна выводить на экран результаты навигации и информацию о препятствиях.

3. Расширение программы для работы с несколькими датчиками и дронами: Измените программу таким образом, чтобы она могла работать с несколькими датчиками и датчиками- обнаружителями одновременно. Добавьте возможность выбора датчика и передачу соответствующего идентификатора через API. Реализуйте также функциональность для работы с несколькими дронами. Добавьте возможность выбора дрона и передачу соответствующего идентификатора через API. Программа должна выводить на экран результаты навигации и информацию о препятствиях для выбранного датчика и дрона.

### ***Задание к теме 1.6***

*Реализация системы распознавания и классификации объектов с помощью машинного обучения на дроне (4 ч.)*

1. Сбор и разметка данных: Соберите набор данных, содержащий изображения объектов, которые вы хотите распознавать и классифицировать с помощью машинного обучения. Разметьте каждое изображение, указав соответствующий класс объекта. Необходимо иметь достаточное количество разнообразных изображений для обучения модели.

2. Обучение модели: Используйте выбранный фреймворк машинного обучения, например TensorFlow или PyTorch, для обучения модели классификации объектов. Разделите набор данных на обучающую и проверочную выборки. Примените выбранный алгоритм обучения, такой как сверточные нейронные сети (CNN), и настройте параметры модели для достижения оптимальной производительности. Обучите модель на обучающей выборке и проверьте ее точность на проверочной выборке.

3. Реализация на дроне: Интегрируйте обученную модель классификации объектов на дроне. Обеспечьте передачу видеопотока с камеры дрона на компьютер или устройство, на котором будет запущена модель. Примените модель к каждому кадру видеопотока для распознавания и классификации объектов. Результаты классификации могут быть отображены на экране или переданы на другое устройство для дальнейшей обработки или действий.

4. Тестирование и оптимизация: Проведите тестирование системы распознавания и классификации объектов на дроне. Оцените производительность модели на различных типах объектов и в различных условиях освещения. Выявите возможные проблемы и ошибки, такие как ложные срабатывания или низкую точность классификации, и проведите оптимизацию модели и алгоритма для улучшения результатов. Проверьте работу системы в реальных условиях, включая полеты дрона и классификацию объектов в реальном времени.

### ***Задание к теме 1.7***

*Оптимизация производительности беспилотных летательных аппаратов с использованием асинхронного программирования и многопоточности (3 ч.)*

1. Анализ производительности: Проанализируйте производительность вашего беспилотного летательного аппарата и выявите узкие места, где можно применить асинхронное программирование и многопоточность для оптимизации. Рассмотрите операции, которые занимают значительное время выполнения или вызывают блокировки, и определите, какие из них можно выполнить асинхронно или параллельно.

2. **Разделение задач:** Разделите задачи вашего беспилотного летательного аппарата на независимые части, которые могут выполняться асинхронно или параллельно. Некоторые примеры таких задач могут включать считывание данных с датчиков, обработку видеопотока, анализ обстановки, планирование маршрута и управление двигателями.
3. **Реализация асинхронности и многопоточности:** Используя выбранный язык программирования (например, Python), внедрите асинхронное программирование и многопоточность в код вашего беспилотного летательного аппарата. Для асинхронного программирования можно использовать конструкции, такие как асинхронные функции, корутины и event loop. Для многопоточности можно использовать потоки выполнения, пулы потоков или другие средства, предоставляемые языком программирования.
4. **Обработка данных:** Оптимизируйте обработку данных, используя асинхронные операции или многопоточность. Например, вы можете считывать данные с датчиков асинхронно, чтобы минимизировать блокировки и задержки. Также можно выполнять параллельную обработку данных, например, анализируя видеопоток с нескольких камер с использованием многопоточности.

### ***Задание к теме 1.8***

*Разработка системы планирования миссий для беспилотных летательных аппаратов с учетом ограничений и целей (3 ч.)*

1. **Определение ограничений и целей:** Определите ограничения и цели вашей системы планирования миссий для беспилотных летательных аппаратов. Ограничения могут включать ограничения по времени полета, ограничения по расстоянию, ограничения по высоте и другие физические или законодательные ограничения. Цели могут включать достижение определенных мест или точек интереса, выполнение задач наблюдения, съемки или доставки.
2. **Анализ требований:** Проанализируйте требования к системе планирования миссий. Определите, какие данные и информация необходимы для планирования миссий, такие как карты, информация о препятствиях, местоположение целей и ограничения. Также определите методы и алгоритмы планирования, которые могут быть применены для достижения целей при соблюдении ограничений.
3. **Реализация системы планирования:** Используя выбранный язык программирования (например, Python), реализуйте систему планирования миссий для беспилотных летательных аппаратов. Разработайте алгоритмы планирования, которые учитывают ограничения и цели, и определите соответствующие структуры данных и функции для реализации системы. Интегрируйте необходимые данные, такие как карты и информацию о препятствиях, для принятия во внимание окружающую среду при планировании миссий.

### ***Задание к теме 2.1***

*Обзор продвинутой технологий беспилотных летательных аппаратов (1 ч.)*

1. **Исследуйте продвинутые технологии,** используемые в беспилотных летательных аппаратах. Опишите основные компоненты, которые используются в современных беспилотных летательных аппаратах, такие как автопилоты, датчики, системы распознавания образов, системы коммуникаций и т.д.
2. **Выберите одну из технологий,** которые вы изучили, и опишите ее подробнее. Рассмотрите основные принципы работы, преимущества и недостатки, а также примеры использования в реальных ситуациях.

3. Предложите свой собственный проект, используя выбранную технологию. Опишите, как вы будете использовать эту технологию в своем проекте, какие компоненты и инструменты вам понадобятся, и как вы будете проверять работу своего проекта. Подумайте о том, как вы можете улучшить или расширить свой проект в будущем, и как он может быть полезен в реальном мире.

### **Задание к теме 2.2**

*Работа с датчиками и сенсорами беспилотных летательных аппаратов (2 ч.)*

1. Изучите различные типы датчиков и сенсоров, которые могут использоваться в беспилотных летательных аппаратах, такие как GPS, гироскопы, акселерометры, магнитометры, датчики давления, камеры и многие другие. Опишите, как каждый датчик работает, какие данные он собирает, и как эти данные могут быть использованы в беспилотных летательных аппаратах.

2. Выберите один тип датчика или сенсора и опишите, как он может быть использован в конкретной ситуации, связанной с беспилотными летательными аппаратами. Рассмотрите преимущества и недостатки использования этого датчика или сенсора, а также возможные ограничения и сложности в работе с ним.

3. Разработайте свой собственный проект, используя выбранный датчик или сенсор. Опишите, как вы будете использовать этот датчик или сенсор в своем проекте, какие компоненты и инструменты вам понадобятся, и как вы будете проверять работу своего проекта. Подумайте о том, как вы можете улучшить или расширить свой проект в будущем, и как он может быть полезен в реальном мире.

### **Задание к теме 2.3**

*Обработка видео и изображений с помощью Python для беспилотных летательных аппаратов (2 ч.)*

1. Изучите различные методы обработки видео и изображений, которые могут использоваться в беспилотных летательных аппаратах, такие как детектирование объектов, распознавание лиц, сегментация изображений и многие другие. Опишите, как каждый метод работает, какие данные он собирает, и как эти данные могут быть использованы в беспилотных летательных аппаратах.

2. Выберите один метод обработки видео или изображений и опишите, как он может быть использован в конкретной ситуации, связанной с беспилотными летательными аппаратами. Рассмотрите преимущества и недостатки использования этого метода, а также возможные ограничения и сложности в работе с ним.

3. Разработайте свой собственный проект, используя выбранный метод обработки видео или изображений. Опишите, как вы будете использовать этот метод в своем проекте, какие компоненты и инструменты вам понадобятся, и как вы будете проверять работу своего проекта. Подумайте о том, как вы можете улучшить или расширить свой проект в будущем, и как он может быть полезен в реальном мире.

### **Задание к теме 2.4**

*Работа с базами данных для хранения данных беспилотных летательных аппаратов (2 ч.)*

1. Изучите различные типы баз данных, которые могут использоваться для хранения данных беспилотных летательных аппаратов, такие как SQL и NoSQL базы данных. Опишите, как каждый тип базы данных работает, какие преимущества и недостатки у каждого типа, и как выбрать подходящую базу данных для конкретной ситуации.
2. Создайте базу данных для хранения данных беспилотных летательных аппаратов, используя выбранный вами тип базы данных. Опишите, как вы будете организовывать данные в базе данных, какие таблицы и поля вы будете использовать, и какие индексы вы будете создавать для быстрого доступа к данным.
3. Разработайте программу на Python для работы с базой данных беспилотных летательных аппаратов. Опишите, как программа будет подключаться к базе данных, какие запросы она будет отправлять для получения и обновления данных, и как она будет обрабатывать ошибки и исключения. Подумайте о том, как вы можете улучшить свою программу в будущем, например, добавив поддержку новых типов данных или интегрируя ее с другими системами.

### **Задание к теме 2.5**

*Разработка алгоритмов и программ для автоматического пилотирования беспилотных летательных аппаратов (2 ч.)*

1. Напишите алгоритм для автоматического пилотирования беспилотного летательного аппарата в режиме «следования за объектом». Какие датчики и аппараты понадобятся для реализации такого алгоритма?
2. Разработайте программу на языке Python для автоматического взлета и посадки беспилотного летательного аппарата. Какие алгоритмы и библиотеки Python можно использовать для решения этой задачи?
3. Как можно оптимизировать работу автоматического пилота беспилотного летательного аппарата с помощью машинного обучения? Какие данные необходимо собирать для обучения модели? Какие алгоритмы машинного обучения можно использовать для решения этой задачи?

### **Задание к теме 2.6**

*Использование машинного обучения и нейронных сетей в беспилотных летательных аппаратах (3 ч.)*

1. В чем заключается процесс обучения нейронной сети в контексте беспилотных летательных аппаратов? Какие типы данных и алгоритмы машинного обучения можно использовать при разработке беспилотных летательных аппаратов?
2. Как можно использовать машинное обучение для обработки сигналов датчиков и улучшения стабильности полета беспилотных летательных аппаратов? Какие типы нейронных сетей могут быть использованы в этом контексте?
3. Как машинное обучение может быть использовано для анализа изображений и видео, полученных с бортовых камер беспилотных летательных аппаратов? Какие типы нейронных сетей используются для обработки изображений и видео, и какие применения этих сетей наиболее популярны в беспилотной авиации?

### **Задание к теме 2.7**

*Разработка и настройка систем управления беспилотными летательными аппаратами (2 ч.)*

1. Что такое система управления беспилотным летательным аппаратом и как она работает?
2. Какие параметры необходимо учитывать при разработке системы управления беспилотным летательным аппаратом и как они влияют на его работу?
3. Какие существуют подходы и методы настройки системы управления беспилотным летательным аппаратом и как выбрать подходящий метод для конкретного случая?

### **Задание к теме 2.8**

*Создание пользовательских интерфейсов для управления беспилотными летательными аппаратами (2 ч.)*

1. Какие компоненты входят в пользовательский интерфейс БПЛА? Какие бывают виды элементов управления и для чего они нужны?
2. Какие технологии можно использовать для создания пользовательского интерфейса БПЛА на языке Python? Назовите несколько библиотек или фреймворков и опишите их особенности.
3. Расскажите о том, какие основные задачи могут решаться с помощью пользовательского интерфейса БПЛА. Какие виды информации должны отображаться на экране управления, чтобы пользователь мог эффективно контролировать БПЛА?

### **Задание к теме 2.9**

*Проект: Разработка программы для автоматического пилотирования беспилотного летательного аппарата на Python (2 ч.)*

1. Определение требований: Опишите основные требования к программе для автоматического пилотирования беспилотного летательного аппарата. Какие функции должна выполнять программа? Какие параметры управления должны быть доступны для настройки? Какие данные необходимо собирать и анализировать?
2. Проектирование программы: Определите основные компоненты программы для автоматического пилотирования беспилотного летательного аппарата. Какие модули Python будут использоваться для управления беспилотным летательным аппаратом? Как будут обрабатываться данные, полученные от датчиков и камер? Как будет происходить анализ данных и выработка решений на основе полученных результатов?
3. Реализация программы: Опишите процесс реализации программы для автоматического пилотирования беспилотного летательного аппарата на Python. Какие инструменты разработки будут использоваться? Как будет происходить тестирование и отладка программы? Какие дополнительные функции могут быть добавлены для улучшения работы программы?

### **Задание к теме 2.10**

*Проект: Разработка системы управления беспилотным летательным аппаратом на Python (2 ч.)*

1. Какие датчики и сенсоры необходимо использовать для управления беспилотным летательным аппаратом и почему? Опишите их работу и возможности.

2. Какие алгоритмы и методы машинного обучения могут быть использованы для управления беспилотным летательным аппаратом? Объясните, как они работают и какие данные необходимы для их обучения.
3. Какие модули Python и библиотеки следует использовать при разработке системы управления беспилотным летательным аппаратом? Обоснуйте выбор каждого из них и опишите, как они будут использоваться в проекте.

### ***Задание к теме 3.1***

*Разработка программного обеспечения для беспилотных летательных аппаратов на основе архитектур ROS и MAVLink (2 ч.)*

1. Объясните, что такое ROS и какова его роль в разработке программного обеспечения для беспилотных летательных аппаратов.
2. Какую роль играет MAVLink в разработке программного обеспечения для беспилотных летательных аппаратов и как он связан с ROS?
3. Опишите процесс разработки программного обеспечения для беспилотных летательных аппаратов на основе архитектур ROS и MAVLink. Какие шаги нужно предпринять и какие инструменты использовать для успешной разработки?

### ***Задание к теме 3.2***

*Разработка и использование дополнительных средств связи между беспилотными летательными аппаратами и главным управляющим центром (1 ч.)*

1. Написать программу на Python, которая будет отправлять информацию с борта беспилотного летательного аппарата на главный управляющий центр через GSM модем.
2. Настроить прием информации с беспилотного летательного аппарата на главном управляющем центре с использованием GSM модема.
3. Исследовать возможность использования спутниковых систем связи (например, Iridium) для передачи данных с беспилотного летательного аппарата на главный управляющий центр и разработать простой прототип на Python для тестирования связи.

### ***Задание к теме 3.3***

*Разработка и применение технологий машинного обучения и искусственного интеллекта для беспилотных летательных аппаратов (2 ч.)*

1. Обзор технологий машинного обучения и искусственного интеллекта, применяемых в беспилотных летательных аппаратах. В задании нужно описать различные виды алгоритмов машинного обучения и искусственного интеллекта, используемые для разработки беспилотных летательных аппаратов, их достоинства и недостатки, а также примеры их применения в реальных проектах.
2. Разработка программного обеспечения для беспилотных летательных аппаратов с использованием технологий машинного обучения и искусственного интеллекта. Здесь необходимо описать процесс разработки и реализации алгоритмов машинного обучения и искусственного интеллекта, а также их интеграцию в программное обеспечение беспилотных летательных аппаратов. Можно также упомянуть о необходимых библиотеках и инструментах для работы с машинным обучением и искусственным интеллектом в Python.

3. Тестирование и анализ работы разработанной системы. В этом задании нужно описать методы тестирования и анализа работы системы управления беспилотным летательным аппаратом с использованием машинного обучения и искусственного интеллекта. Необходимо также описать процесс анализа результатов и выявления проблем в работе системы.

### **Задание к теме 3.5**

*Применение компьютерного зрения и обработки изображений для улучшения функциональности беспилотных летательных аппаратов (2 ч.)*

1. Каким образом компьютерное зрение может помочь беспилотным летательным аппаратам? Напишите алгоритм обнаружения объекта на изображении с помощью компьютерного зрения, который может быть использован в беспилотных летательных аппаратах.
2. Рассмотрите возможности использования обработки изображений для определения местоположения и ориентации беспилотных летательных аппаратов. Напишите алгоритм определения ориентации беспилотного летательного аппарата по видеопотоку с его камеры.
3. Рассмотрите задачу распознавания знаков на изображениях и видеопотоках в контексте беспилотных летательных аппаратов. Напишите алгоритм распознавания знака стоп на изображении с помощью компьютерного зрения и определите, как такой алгоритм может быть использован для управления беспилотным летательным аппаратом.

### **Задание к теме 3.6**

*Разработка и применение системы навигации и автоматического пилотирования на основе датчиков GPS и инерциальной навигации (3 ч.)*

1. Объясните, как работает система навигации и автоматического пилотирования на основе датчиков GPS и инерциальной навигации?
2. Какие особенности присущи каждому из датчиков и как они работают в совокупности для обеспечения точной навигации и пилотирования беспилотного летательного аппарата?
3. Какие меры безопасности необходимо принимать при разработке и применении системы навигации и автоматического пилотирования на основе датчиков GPS и инерциальной навигации?

### **Задание к теме 3.7**

*Разработка системы обнаружения и предотвращения столкновений для беспилотных летательных аппаратов (3 ч.)*

1. Определение требований к системе обнаружения и предотвращения столкновений беспилотных летательных аппаратов, включая диапазон дистанции, точность и скорость обработки данных.
2. Изучение алгоритмов компьютерного зрения и машинного обучения для обнаружения объектов и прогнозирования их траекторий.
3. Разработка и тестирование системы обнаружения и предотвращения столкновений на беспилотном летательном аппарате с использованием датчиков и камер, включая анализ полученных данных и оценку эффективности системы в различных условиях.

### **Задание к теме 3.8**

*Проект: Разработка приложения для управления беспилотным летательным аппаратом на основе архитектуры ROS и MAVLink (2 ч.)*

1. Настройка и конфигурация: необходимо настроить и сконфигурировать систему ROS и MAVLink, чтобы они могли работать вместе и обмениваться сообщениями и командами.
2. Разработка пользовательского интерфейса: необходимо разработать пользовательский интерфейс для управления беспилотным летательным аппаратом. Интерфейс должен быть интуитивно понятным и позволять пользователю задавать путь полета, устанавливать скорость и высоту, а также просматривать информацию о состоянии батареи и другие данные.
3. Интеграция с системой управления беспилотным летательным аппаратом: необходимо интегрировать приложение с системой управления беспилотным летательным аппаратом, чтобы приложение могло отправлять команды на управление беспилотным летательным аппаратом и получать данные от него. Также нужно обеспечить возможность автоматического пилотирования по заданному маршруту.

### ***Задание к теме 3.9***

*Разработка системы обнаружения и предотвращения столкновений для беспилотных летательных аппаратов (2 ч.)*

1. Разработать алгоритм обнаружения столкновений на основе данных с датчиков аппарата, таких как LIDAR или радар. Какие параметры и характеристики необходимо учитывать для определения расстояния до других объектов, их скорости и направления движения?
2. Интегрировать систему обнаружения столкновений с системой автоматического пилотирования. Каким образом система автоматического пилотирования должна реагировать на определенные ситуации обнаружения столкновений, и какие меры должны быть приняты для предотвращения столкновений?
3. Оценить эффективность разработанной системы обнаружения и предотвращения столкновений, используя различные тестовые сценарии. Какие метрики и критерии эффективности должны быть использованы для оценки работы системы, и какие дополнительные улучшения могут быть внесены в систему для повышения ее эффективности?

### ***Задание к теме 4.1***

*Основы разработки беспилотных летательных аппаратов на языке Python (2 ч.)*

1. Что такое беспилотный летательный аппарат и какие компоненты входят в его систему управления?
2. Какие типы сенсоров используются для сбора данных на беспилотных летательных аппаратах и как они помогают в автономном пилотировании?
3. Какие базовые алгоритмы используются для автоматического пилотирования беспилотных летательных аппаратов и как они реализуются на языке Python?

### ***Задание к теме 4.2***

*Работа с фреймворками для разработки беспилотных летательных аппаратов на Python (2 ч.)*

1. Выберите один из фреймворков для разработки беспилотных летательных аппаратов на Python (например, PX4 или ArduPilot) и опишите его основные преимущества и недостатки.

2. Расскажите, какие инструменты предоставляет выбранный вами фреймворк для разработки, тестирования и отладки беспилотных летательных аппаратов на Python.
3. Опишите процесс создания простейшего скрипта на Python с использованием выбранного вами фреймворка, который бы позволял беспилотному летательному аппарату пролететь вдоль прямой линии на определенной высоте.

#### ***Задание к теме 4.3***

*Создание беспилотного летательного аппарата на Python: от идеи до конечного продукта (3 ч.)*

1. Определить функциональные требования к беспилотному летательному аппарату, включающие в себя его основные характеристики, способность к автоматическому управлению и передаче информации.
2. Разработать архитектуру беспилотного летательного аппарата на языке Python, включая модули для работы с датчиками, системой навигации, автоматическим пилотом и системой обработки изображений.
3. Протестировать работу беспилотного летательного аппарата в различных условиях, включая проверку его стабильности, скорости и точности при выполнении задач автоматического пилотирования и системы обнаружения препятствий.

#### ***Задание к теме 4.4***

*Разработка системы управления полетом беспилотного летательного аппарата на Python (2 ч.)*

1. Определить основные требования к системе управления полетом: это может быть, например, стабильный полет на определенной высоте и скорости, автоматическая навигация с помощью GPS, обеспечение безопасности полета и т.д.
2. Разработать алгоритмы управления полетом на языке Python: в зависимости от требований, это может быть написание кода для управления двигателями, автопилотом, системой навигации и другими компонентами.
3. Протестировать и отладить систему: после написания кода необходимо провести тестирование системы на различных условиях и отладить возможные ошибки и неполадки.

#### ***Задание к теме 4.5***

*Разработка системы навигации для беспилотного летательного аппарата на Python (3 ч.)*

1. Разработать алгоритм определения местоположения беспилотного летательного аппарата на основе GPS- данных. Для этого нужно исследовать доступные библиотеки для работы с GPS- данными на языке Python, например, библиотеку `pygps`.
2. Разработать алгоритм определения ориентации беспилотного летательного аппарата на основе данных, полученных от инерциальных датчиков. Для этого нужно исследовать доступные библиотеки для работы с инерциальными датчиками на языке Python, например, библиотеку `pyIMU`.

3. Реализовать алгоритм слияния данных о местоположении и ориентации беспилотного летательного аппарата с помощью фильтра Калмана. Для этого нужно изучить теорию фильтра Калмана и реализовать его на языке Python с использованием полученных ранее данных о местоположении и ориентации.

### **Задание к теме 4.6**

*Создание пользовательского интерфейса для управления беспилотным летательным аппаратом на Python (3 ч.)*

#### **Пошаговый tutorial: Управление беспилотным летательным аппаратом**

*Напишите программу, которая представляет собой управление беспилотным летательным аппаратом с помощью графического интерфейса. Ваша задача состоит в том, чтобы управлять беспилотником с помощью клавиш со стрелками на клавиатуре и достигать красных целей на поле, представленных в виде кружочков.*

#### **Требования:**

1. Запустите программу.
  2. Появится графическое окно с полем и белым треугольником (беспилотным аппаратом).
  3. Используя клавиши со стрелками на клавиатуре, перемещайте беспилотный аппарат по полю.
  4. Ваша цель - достичь красные кружочки на поле.
  5. Чтобы достичь красный кружочек, щелкните левой кнопкой мыши на треугольнике, чтобы беспилотный аппарат двигался в направлении к кружочку.
  6. Когда беспилотный аппарат достигает красного кружочка, цвет кружочка изменяется на зеленый, и в консоль выводится сообщение «Беспилотник достиг цели X», где X - номер цели.
  7. Продолжайте перемещать беспилотный аппарат, достигая остальные красные кружочки.
- Попробуйте достичь все красные кружочки на поле с помощью управления беспилотным аппаратом. Удачи!

#### **Пример кода**

Скачать пример кода вы можете перейдя по ссылке: <https://disk.yandex.ru/d/Y6xq2UKQOIPpLw>

#### **Подробный разбор кода**

```
import tkinter as tk
Импортирование библиотеки tkinter.
window = tk.Tk()
window.title(«Управление беспилотным летательным аппаратом»)
window.geometry(«800x600»)
```

Создание графического окна.

```
canvas = tk.Canvas(window, bg=«black», width=800, height=600)  
canvas.pack()
```

Создание Canvas (поле для отображения графических элементов).

```
triangle = canvas.create_polygon(400, 300, 350, 350, 450, 350, fill=«white»)
```

Создание белого треугольника, представляющего беспилотник.

```
locations = [(200, 200), (400, 400), (600, 200)]
```

```
circle_radius = 10
```

```
circle_color = «red»
```

```
circle_objects = []
```

```
for location in locations:
```

```
    x, y = location
```

```
    circle = canvas.create_oval(x - circle_radius, y - circle_radius, x + circle_radius, y + circle_radius, fill=circle_color)
```

```
    circle_objects.append(circle)
```

Создание списка с координатами красных кружочков (локаций) и их отображение на поле.

```
def reach_location(circle):
```

```
    canvas.itemconfigure(circle, fill=«green»)
```

```
    index = circle_objects.index(circle)
```

```
    message = f«Беспилотник достиг цели {index+1}»
```

```
    print(message)
```

Определение функции reach\_location, которая изменяет цвет кружочка на зеленый и выводит сообщение о достижении цели.

```
def move_triangle(event):
```

```
    key = event.keysym
```

```
    coords = canvas.coords(triangle)
```

```
    if key == «Up»:
```

```
        canvas.move(triangle, 0, - 10)
```

```

elif key == «Down»:
    canvas.move(triangle, 0, 10)
elif key == «Left»:
    canvas.move(triangle, - 10, 0)
elif key == «Right»:
    canvas.move(triangle, 10, 0)

```

**check\_reached\_location()**

Определение функции move\_triangle, которая перемещает треугольник (беспилотник) при нажатии стрелок клавиатуры.

```

def handle_click(event):

```

```

    x, y = event.x, event.y

```

```

    item = canvas.find_closest(x, y)[0]

```

```

if item == triangle:

```

```

    closest_circle = None

```

```

    closest_distance = float(«inf»)

```

```

for circle in circle_objects:

```

```

    circle_coords = canvas.coords(circle)

```

```

    circle_x, circle_y = circle_coords[0], circle_coords[1]

```

```

    distance = ((circle_x - x) ** 2 + (circle_y - y) ** 2) ** 0.5

```

```

    if distance < closest_distance:

```

```

        closest_circle = circle

```

```

        closest_distance = distance

```

```

if closest_circle and canvas.itemcget(closest_circle, «fill») == circle_color:

```

```

    reach_location(closest_circle)

```

Определение функции handle\_click, которая обрабатывает клик мыши и определяет ближайший кружочек к беспилотнику.

```

def check_reached_location():

```

```

    triangle_coords = canvas.coords(triangle)

```

```

    triangle_x, triangle_y = triangle_coords[0], triangle_coords[1]

```

```

    for circle in circle_objects:

```

```

        circle_coords = canvas.coords(circle)

```

```

        circle_x, circle_y = circle_coords[0], circle_coords[1]

```

```
if (circle_x - circle_radius <= triangle_x <= circle_x + circle_radius) and (circle_y - circle_radius <= triangle_y <= circle_y + circle_radius)
and canvas.itemcget(circle, «fill») == circle_color:
    reach_location(circle)
```

Определение функции `check_reached_location`, которая проверяет, достиг ли беспилотник какой-либо кружочек.

```
window.bind(«<Key>», move_triangle)
canvas.bind(«<Button-1>», handle_click)
window.focus_set()
```

Привязка функций к событиям (нажатия клавиш и клики мыши).

```
window.mainloop()
```

Запуск главного цикла окна.

**Для создания проекта «Управление беспилотным летательным аппаратом» с использованием графического интерфейса на основе библиотеки Tkinter, вам потребуется знать следующие вещи:**

1. Базовые знания языка программирования Python: Чтобы понимать синтаксис и основные концепции Python, необходимые для создания программы.
2. Установка и импорт библиотеки Tkinter: Tkinter - стандартная библиотека для создания графических интерфейсов в Python. Убедитесь, что у вас установлен Python, включающий библиотеку Tkinter, или установите ее отдельно. Затем вы можете импортировать ее в свой код с помощью команды `import tkinter as tk`.
3. Работа с окном и холстом (Canvas): Создание главного окна приложения с помощью `Tk()`, задание его свойств (название, размер) и создание холста с помощью `Canvas()`, на котором будут располагаться графические элементы.
4. Создание графических элементов: Используя методы `create_polygon()` и `create_oval()` холста, вы можете создать треугольник (беспилотный аппарат) и круги (цели).
5. Обработка событий: Привязка функций-обработчиков к событиям, таким как нажатие клавиш или щелчок мыши. Ваш код должен реагировать на события и выполнять соответствующие действия, например, перемещать беспилотный аппарат или проверять достижение целей.
6. Работа с элементами холста: Использование методов `move()`, `itemconfigure()` и `coords()` для перемещения беспилотного аппарата, изменения цвета кружочков и получения координат элементов.
7. Организация логики программы: Реализация функций, которые проверяют достижение целей, выводят сообщения и обновляют состояние графических элементов.

8. Главный цикл окна: Запуск главного цикла с помощью метода `mainloop()`, который обрабатывает события и обновляет интерфейс приложения.

Знание этих основных концепций и методов позволит вам создать и запустить программу «Управление беспилотным летательным аппаратом». Вы также можете обратиться к документации Tkinter и примерам кода для более подробной информации и расширения функциональности вашего проекта.

Календарный график:

Первый модуль с 01.10.2023 - 30.11.2023

Второй модуль: с 01.12.2023 - 31.01.2024

Третий модуль: с 01.02.2024 - 31.03.2024

Четвертый модуль: с 01.04.2024 - 31.05.2024

*Занятия проводятся 2- 3 раза в неделю по 2 академических часа, длительность одного академического часа – 45 минут.*

1 занятие 2 академических часа

#### Календарно- тематическое планирование

№	Тема и № модуля	Тема занятия	Кол- во занятий*	Кол- во часов	Дата
1	<b>Модуль 1: Продвинутое программирование и управление беспилотными летательными аппаратами с использованием Python.</b>	Тема 1.1 Алгоритмы планирования траекторий для беспилотных летательных аппаратов.	1	1	2.10.23
2		Тема 1.2 Реализация алгоритмов машинного обучения для автономного управления дронами	1	1	2.10.23
3		Тема 1.3 Работа с компьютерным зрением и обработка изображений в контексте беспилотных летательных аппаратов	1	1	6.10.23
4		Тема 1.4 Разработка системы оптической стабилизации камеры на дроне	3	6	6.10.23 9.10.23 11.10.23

					16.10.23
5		Тема 1.5 Использование датчиков и датчиков- обнаружителей (например, GPS, инфракрасные сенсоры, радары) для автономной навигации и избегания препятствий	3	6	16.10.23 20.10.23 23.10.23 27.10.23
6		Тема 1.6 Реализация системы распознавания и классификации объектов с помощью машинного обучения на дроне	2	3	27.10.23 7.11.23
7		Тема 1.7 Оптимизация производительности беспилотных летательных аппаратов с использованием асинхронного программирования и многопоточности	2	4	13.11.23 16.11.23
8		Тема 1.8 Разработка системы планирования миссий для беспилотных летательных аппаратов с учетом ограничений и целей	2	4	20.11.23 23.11.23
9		Тема 1.9 Интеграция беспилотных летательных аппаратов с облачными сервисами для передачи данных, хранения информации и управления дронами из удаленного местоположения	2	4	27.11.23 29.11.23
		Промежуточная аттестация	1	1	30.11.23
10	<b>Модуль 2: Продвинутые техники программирования беспилотных летательных аппаратов на языке Python</b>	Тема 2.1 Обзор продвинутых технологий беспилотных летательных аппаратов	1	2	1.12.23
11		Тема 2.2 Работа с датчиками и сенсорами беспилотных летательных аппаратов	2	3	4.12.23 8.12.23
12		Тема 2.3 Обработка видео и изображений с помощью Python	2	3	8.12.23

		для беспилотных летательных аппаратов			11.12.23
13		Тема 2.4 Работа с базами данных для хранения данных беспилотных летательных аппаратов	2	3	14.12.23 18.12.23
14		Тема 2.5 Разработка алгоритмов и программ для автоматического пилотирования беспилотных летательных аппаратов	2	3	18.12.23 22.12.23
15		Тема 2.6 Использование машинного обучения и нейронных сетей в беспилотных летательных аппаратах	2	4	25.12.23 27.12.23
16		Тема 2.7 Разработка и настройка систем управления беспилотными летательными аппаратами	2	3	9.01.24 12.01.24
17		Тема 2.8 Создание пользовательских интерфейсов для управления беспилотными летательными аппаратами	2	3	12.01- 24 18.01.24
18		Тема 2.9 Проект: Разработка программы для автоматического пилотирования беспилотного летательного аппарата на Python	2	3	22.01.24 25.01.24
19		Тема 2.10 Проект: Разработка системы управления беспилотным летательным аппаратом на Python	2	3	25.01.24 29.01.24
		Промежуточная аттестация	1	1	30.01.24
20	<b>Модуль 3: Расширенные технологии и тенденции программирования беспилотных летательных аппаратов</b>	Тема 3.1 Разработка программного обеспечения для беспилотных летательных аппаратов на основе архитектур ROS и MAVLink	2	3	1.02.24 3.02.24
21		Тема 3.2 Разработка и использование дополнительных средств связи между беспилотными летательными аппаратами и главным управляющим центром	2	3	3.02.24 8.02.24

22	<b>на языке Python</b>	Тема 3.3 Разработка и применение технологий машинного обучения и искусственного интеллекта для беспилотных летательных аппаратов	2	3	9.02.24 12.02.24
23		Тема 3.4 Интеграция различных датчиков и устройств для расширения функциональности беспилотных летательных аппаратов	2	3	12.02.24 19.02.24
24		Тема 3.5 Применение компьютерного зрения и обработки изображений для улучшения функциональности беспилотных летательных аппаратов	2	3	21.02.24 24.02.24
25		Тема 3.6 Разработка и применение системы навигации и автоматического пилотирования на основе датчиков GPS и инерциальной навигации	2	4	24.02.24 29.02.24 1.03.24
26		Тема 3.7 Разработка системы обнаружения и предотвращения столкновений для беспилотных летательных аппаратов	2	4	1.03.24 4.03.24 7.03.24
27		Тема 3.8 Проект: Разработка приложения для управления беспилотным летательным аппаратом на основе архитектуры ROS и MAVLink	2	3	7.03.24 11.03.24
28		Тема 3.9 Проект: Разработка системы обнаружения и предотвращения столкновений для беспилотных летательных аппаратов	2	4	18.03.24 21.03.24
		Промежуточная аттестация	1	1	23.03.24
29		<b>Модуль 4: Практические примеры разработки беспилотных</b>	Тема 4.1 Основы разработки беспилотных летательных аппаратов на языке Python	2	4
30	Тема 4.2 Работа с фреймворками для разработки беспилотных летательных аппаратов на Python		2	4	8.04.24

	<b>летательных аппаратов на языке Python</b>				11.04.24
<b>31</b>		Тема 4.3 Создание беспилотного летательного аппарата на Python: от идеи до конечного продукта	2	4	15.04.24 18.04.24
<b>32</b>		Тема 4.4 Разработка системы управления полетом беспилотного летательного аппарата на Python	2	3	22.04.24 25.04.24
<b>33</b>		Тема 4.5 Разработка системы навигации для беспилотного летательного аппарата на Python	2	4	25.04.24 27.04.24 6.05.24
<b>34</b>		Тема 4.6 Создание пользовательского интерфейса для управления беспилотным летательным аппаратом на Python	3	5	6.05.24 10.05.24 13.05.24
<b>35</b>		Тема 4.7 Продвинутое техники разработки беспилотных летательных аппаратов на Python: машинное обучение, нейронные сети, искусственный интеллект	2	4	16.05.24 20.05.24
<b>36</b>		Консультация	1	2	23.05.24
		Промежуточная аттестация	1	1	25.05.24

**\*количество занятий не включает часы, отведенные на самостоятельное изучение, и часы, отведенные на прохождение аттестации**

### Учебно- методические материалы

Наименование поля	Допустимые значения полей	Значение полей	Значение полей	Значение полей	Значение полей
Порядковый номер модуля	строка не менее 10 символов	1	2	3	4
Методы, формы и технологии	строка не менее 10 символов	Для проведения синхронных занятий применяется программа видеоконференцсвязи Zoom. Дополнительно для организации работы слушателей могут использоваться Google- сервисы. Асинхронная работа слушателей реализуется на базе электронного курса на платформе электронного обучения LMS Odin.			
Методические разработки	строка не менее 10 символов	Программа реализуется в формате смешанного обучения. Синхронные занятия включают интерактивные лекции и практические занятия, предусматривающие групповую и индивидуальную работу слушателей. Для организации асинхронной работы слушателей используются записи синхронных занятий, презентации, конспекты лекций, практические задания и тестовые вопросы, размещаемые в электронном курсе на платформе электронного обучения LMS Odin.			
Материалы модуля	строка не менее 10 символов	Для изучения программы используется электронный курс в LMS Odin. Электронный курс включает информационно- организационные материалы по программе (учебно- тематический план, календарный график работы по программе, информацию о преподавателях программы, чат для объявлений и вопросов, сведения о результатах обучения), набор видеолекций, презентации к лекциям, набор ссылок на внешние ресурсы, систему заданий с инструкциями, списки основной и дополнительной литературы.			
Учебная литература	строка не менее 10 символов	1. «Python для сложных задач: наука о данных и искусственный интеллект», Марко Гуззи, Эрик Шлутер, Инге Соргер, Джош Мингес, Дэн Велч, Энтони Де Барто, 2019 г.			

		<p>2. «Python и машинное обучение: современное зрение компьютеров», Себастьян Рашка, 2017 г.</p> <p>3. «Глубокое обучение на Python. Керас, TensorFlow, PyTorch», Франсуа Шолле, 2019 г.</p> <p>4. «Учим Python, делая крутые игры», Эрик Мэтиз, 2019 г.</p> <p>5. «Python. Карманный справочник», Марк Лутц, 2018 г.</p>
--	--	---

#### Материально-технические условия реализации программы

Наименование поля	Допустимые значения полей	Значение полей	Значение полей	Значение полей	Значение полей
Порядковый номер модуля	строка не менее 10 символов	1	2	3	4
Наименование требуемого оборудования	строка не менее 2 символов	<p>Для работы на платформе электронного обучения LMS Odin рекомендуется использовать персональный компьютер.</p> <p>Для успешной реализации Программы необходимо, чтобы рабочее место обучающегося и преподавателя включали в себя:</p> <ul style="list-style-type: none"> <li>• компьютеры, обеспечивающие возможность работы с мультимедийным контентом: воспроизведение видеоизображений, качественный стереозвук в наушниках, речевой ввод с микрофона и др.</li> </ul> <p>периферийное оборудование:</p> <ul style="list-style-type: none"> <li>• принтер (черно/белой печати, формата А4);</li> <li>• устройства для ввода визуальной информации (сканер, цифровой фотоаппарат, web- камера и пр.);</li> </ul>			

		<ul style="list-style-type: none"> <li>● устройства создания графической информации (графический планшет), используемые для создания и редактирования графических объектов, ввода рукописного текста;</li> <li>● акустические колонки;</li> <li>● оборудование, обеспечивающее подключение к сети Интернет (комплект оборудования для подключения к сети Интернет, сервер).</li> </ul>
Наименование требуемого программного обеспечения	строка не менее 2 символов	<p>программное обеспечение компьютера:</p> <ul style="list-style-type: none"> <li>● операционная система семейства MacOS или Windows;</li> <li>● программа IDLE, PyCharm</li> <li>● файловый менеджер (в составе операционной системы или др.);</li> <li>● почтовый клиент (в составе операционных систем или др.);</li> <li>● браузер (в составе операционных систем или др.);</li> <li>● мультимедиа проигрыватель (в составе операционной системы или др.);</li> <li>● антивирусная программа;</li> <li>● программа- архиватор;</li> <li>● программа- переводчик;</li> <li>● программа интерактивного общения;</li> <li>● текстовый редактор;</li> <li>● растровый графический редактор;</li> <li>● звуковой редактор;</li> <li>● редактор Web- страниц.</li> </ul>
Электронные информационные ресурсы	строка не менее 10 символов	1. Информационный ресурс по основам БПЛА в свободном доступе от компании ГК Геоскан: <a href="https://docs.geoscan.aero/ru/master/database/base-module/base-main.html">https://docs.geoscan.aero/ru/master/database/base-module/base-main.html</a>

		<p>2. Информационный ресурс по основам программирования на Python:  <a href="https://habr.com/ru/articles/31180/">https://habr.com/ru/articles/31180/</a></p> <p>3. Информационный ресурс по бортовой аппаратуре БПЛА в свободном доступе от компании ГК Геоскан:  <a href="https://docs.geoscan.aero/ru/master/module/module_main.html">https://docs.geoscan.aero/ru/master/module/module_main.html</a></p> <p>4. Информационный ресурс по программированию БПЛА в свободном доступе от компании ГК Геоскан:  <a href="https://docs.geoscan.aero/ru/master/programming/programming.html">https://docs.geoscan.aero/ru/master/programming/programming.html</a></p>
Электронные образовательные ресурсы	строка не менее 10 символов	<p>1. Электронное свободно распространяемое пособие по основам БПЛА:  <a href="https://infourok.ru/bespilotnye-letatelnye-apparaty-bpla-kniga-1-teoriya-4559907.html?ysclid=lj4k4sl1mt287611123">https://infourok.ru/bespilotnye-letatelnye-apparaty-bpla-kniga-1-teoriya-4559907.html?ysclid=lj4k4sl1mt287611123</a></p> <p>2. Электронное свободно распространяемое пособие по основам Python:  <a href="https://elar.urfu.ru/bitstream/10995/28769/1/978-5-7996-1198-9_2014.pdf?ysclid=lj4k928vd8295170590">https://elar.urfu.ru/bitstream/10995/28769/1/978-5-7996-1198-9_2014.pdf?ysclid=lj4k928vd8295170590</a></p> <p>3. Электронное свободно распространяемое пособие по основам БПЛА:  <a href="https://infourok.ru/bespilotnye-letatelnye-apparaty-bpla-kniga-1-teoriya-4559907.html?ysclid=lj4k4sl1mt287611123">https://infourok.ru/bespilotnye-letatelnye-apparaty-bpla-kniga-1-teoriya-4559907.html?ysclid=lj4k4sl1mt287611123</a></p> <p>4. Электронное свободно распространяемое пособие по основам Python:  <a href="https://elar.urfu.ru/bitstream/10995/28769/1/978-5-7996-1198-9_2014.pdf?ysclid=lj4k928vd8295170590">https://elar.urfu.ru/bitstream/10995/28769/1/978-5-7996-1198-9_2014.pdf?ysclid=lj4k928vd8295170590</a></p>

**Адреса и координаты (в случае если дополнительная общеобразовательная программа реализуется посредством сетевой формы реализации образовательных программ (в случае использования очной формы без применения дистанционных технологий))**

№ п/п	Название адрес	Адрес	Код адреса	Долгота	Широта
целое число	строка	строка	Целое число	вещественное число	вещественное число