CuroAI Программирование и обучение искусственному интеллекту

Ответы к рабочим листам и пояснения для педагога

Ответы и пояснения предназначены для педагога. Учащимся выдайте рабочие листы без ответов.

Руководство по использованию:

• Приведены примеры кода, ожидаемые результаты и вопросы для обсуждения каждого занятия.
• ✏ — примеры ответов к заданиям учащихся.
• — пояснения педагогу.
Примечания отмечают параметры, которые нужно подготовить или проверить перед запуском на своём наборе.
• = Направление ответа на вопрос для размышления (руководство, а не окончательные ответы)

Команды сверены с исходниками Python Studio от 11.11.2026. Пропуски в рабочих листах заполняют учащиеся; адреса обученных моделей подставляет педагог. Перед занятием выполните пробу на своём наборе: проверка редакции не заменяет испытание оборудования.

Занятие 1 Вперед и назад

✏ ШАГ 1 Результаты наблюдения
Время движения робота вперед: 1 секунда.
Время движения робота назад: 1 секунда.
sleep(1) приостанавливает выполнение следующих команд на секунду. Уже включённый мотор в это время продолжает работать.

✏ ШАГ 2 Результаты эксперимента
sleep(0.5): мотор работает до следующей команды полсекунды.
sleep(2): мотор работает до следующей команды две секунды.
sleep(3): мотор работает до следующей команды три секунды.
sleep(0.1): короткое действие — одна десятая секунды.

Если увеличить паузу sleep(), мотор будет дольше работать до следующей команды.
Когда значение сна уменьшается → Робот ненадолго движется.

ШАГ 3 Код Ответ:

```
from cubroid import dc_motor, sleep

for i in range(3):    # повторить три раза ◂
    dc_motor.go()
    sleep(1)          # 1 секунда; значение выбирает учащийся ◂
    dc_motor.back()   # движение назад ◂
    sleep(1)          # 1 секунда; значение выбирает учащийся ◂
dc_motor.stop()
```

Педагогу: вместо range(3) допустимы range(5) и другие значения. Длительность пауз sleep дети выбирают самостоятельно. Важно правильно заполнить range() и back().

Примеры ответов для размышления (руководство, а не окончательные ответы)
• Последовательное выполнение: код выполняется по одной строке, сверху вниз.
• Сила одной строки кода: поскольку компьютеры (роботы) точно следуют командам человека, одна

строка кода создает физическое движение в реальном мире.

Занятие 2 «Поворот и управление скоростью»

✏ ШАГ 1 Изучите

go() включает движение обоих моторов вперёд.
right() задаёт поворот направо. Проверьте действие моторов на своей модели.
range(4) задаёт четыре повтора прямого участка и поворота.
speed(50, 200): левый мотор вращается медленнее правого, модель поворачивает влево.

Пример кода ШАГА 2 (Треугольник):

```
from cubroid import dc_motor, sleep

for i in range(3):    # треугольник: три стороны ◂
    dc_motor.go()     # движение вперёд ◂
    sleep(1)
    dc_motor.right()  # поворот направо (120° в замысле; время подобрать) ◂
    sleep(0.7)        # время поворота: подобрать на модели ◂
dc_motor.stop()
```

Совет учителю: рисовать точные треугольники/круги сложно. Ключевым моментом является обнаружение «связи между количеством повторений и временем поворота». Несовершенные формы – это нормально.

Примеры ответов для размышления (руководство, а не окончательные ответы)

• Если левое и правое колёса вращаются с разной скоростью, робот поворачивает. Он отклоняется в сторону более медленного колеса.

• Автомобиль поворачивает передние колёса. В этой модели робота направление меняется за счёт разной скорости левого и правого колёс.

Занятие 3 «Эмоции со светодиодом»

Код ШАГА 2 Ответ:

```
from cubroid import led, sleep

faces = ["happy", "sad", "angry", "surprise", "sleepy"]
for f in faces:
    led.face(f)
    print(f"Выражение: {f}")
    sleep(1)        # пауза 1 секунда; значение выбирает учащийся
led.off()
```

✏ Наблюдательные пункты

Доступные выражения: счастливый, грустный, злой, удивленный, сонный, любовь.
Значки: сердечко, звездочка, галочка и т. д.
Bled.text("HELLO", 0.5), 0.5 = скорость прокрутки.

Педагогу: дети самостоятельно выбирают выражения лица и их порядок в списке. Пять и более вариантов — хороший результат. Длительность пауз sleep также выбирают дети. Главное — понимание списков и циклов for.

Примеры ответов для размышления (руководство, а не окончательные ответы)

Список хранит упорядоченную последовательность элементов. К нужному элементу обращаются по индексу.

Сравните запись семи изображений с отдельными командами и с обходом списка в цикле. Число строк зависит от оформления программы.

Занятие 4 Звук и мелодия

✏ ШАГ 1 Изучите

Пара («sol», 1): нота Соль и длительность в одну долю. Идентификатор ноты в коде сохраняют как в примере.

BPM — число долей в минуту. При темпе 120 одна доля длится полсекунды.

Пример кода для мелодии на этапе 2:

```
from cubroid import sound

sound.melody([
    ("mi", 1), ("re", 1), ("do", 1), ("re", 1),
    ("mi", 1), ("mi", 1), ("mi", 2),
    ("re", 1), ("re", 1), ("re", 2),
    ("mi", 1), ("sol", 1), ("sol", 2),
], bpm=120)
```

Педагогу: дети сочиняют мелодию самостоятельно. Важно, чтобы они понимали структуру кортежа (нота, длительность). Достаточно даже двух тактов.

Занятие 5 RGB-подсветка + датчики

✏ ШАГ 1. Эксперимент с RGB.
RGB(255,0,0) = Красный
RGB(0,255,0) = Зеленый
RGB(0,0,255) = Синий
RGB(255,255,0) = Желтый
RGB(255,0,255) = Фиолетовый (Пурпурный)
RGB(0,255,255) = Небесно-голубой (голубой)

Датчик приближения: предмет близко — is_near() возвращает True; далеко — False.
Датчик касания: при нажатии → is_pressed() = True

Код ШАГА 2 Ответ:

```
from cubroid import dc_motor, proximity_sensor, master, sleep

while True:
    if proximity_sensor.is_near():
        master.red()
        dc_motor.go()
    elif light_touch_sensor.is_pressed():  (условие можно изменить)
        master.blue()
    else:
        master.off()
        dc_motor.stop()
    sleep(0.1)
```

Проверено по API от 11.11.2026: proximity_sensor.is_near() — приближение; light_touch_sensor.is_pressed() — нажатие; light_touch_sensor.brightness() — нормированное показание яркости.

Примеры ответов для размышления (руководство, а не окончательные ответы)

• пример if/elif/else из реальной жизни: «Если идет дождь, возьми зонтик; если идет снег, наденьте перчатки; в противном случае просто выйди'

• Датчик→Процесс→Выход: датчик считывает окружающую среду (вход), код принимает решения (процесс), двигатель/светодиод реагирует (выход).

Занятие 6. Визуализация данных датчиков.

✏ ШАГ 1 Изучите

Ось X: время (непрерывный поток)

Ось Y: значение датчика (яркость, расстояние приближения и т. д.).

Яркость: высокое значение при ярком свете, низкое при темноте (может быть изменено в зависимости от датчика)

Педагогу: дети сами планируют опыт. Например: «Как меняется график датчика приближения, если несколько раз поднести руку и отвести её?» Значения передаются на график внутри цикла while True.

Примеры ответов для размышления (руководство, а не окончательные ответы)

• Просмотр в виде графиков: легче сразу уловить тенденции и закономерности по сравнению с необработанными цифрами.

• while True: бесконечный цикл. Требуется для непрерывного считывания данных датчиков. Внимание: зацикливается навсегда без условия выхода или разрыва.

Занятие 7 Создание функций (def)

✏ ШАГ 1. Концепции.
Функция — именованный фрагмент программы, который можно вызвать повторно.
def: сокращение от «define».
(): Символ, используемый при вызове функции. Параметры можно добавить позже
Определение функции создаёт её тело; вызов запускает команды функции.

Пример кода ШАГА 2:

```
from cubroid import dc_motor, led, sound, sleep

def spin():              # определение функции 1 ◂
    dc_motor.right()
    sleep(0.5)

def wave():              # определение функции 2 ◂
    led.face("happy")
    sound.play("do")
    sleep(0.3)

def bow():               # определение функции 3 ◂
    dc_motor.go()
    sleep(0.3)
    dc_motor.back()
    sleep(0.3)

# Исполни хореографию!
for i in range(2):       ◂
    spin()               # вызов функции ◂
    wave()               # вызов функции ◂
bow()
dc_motor.stop()
```

Педагогу: названия и содержание функций дети выбирают самостоятельно. Важно освоить определение функции с помощью def и её вызов с (). Три и более определения — хороший результат.

Примеры ответов для размышления (руководство, а не окончательные ответы)

Сравните запись нескольких повторяющихся движений отдельно и с вызовами функций.

• Преимущества функций: 1) Повторное использование кода 2) Понимание значения имени 3) Исправление только одного места при изменении

Занятие 8. Создание робота-питомца

Пример ответа по коду:

```
from cubroid import dc_motor, proximity_sensor, led, sound, master, sleep

led.face("happy")
print("Робот-питомец проснулся!")

while True:
    if proximity_sensor.is_near():
        # счастливое состояние
        led.face("happy")
        master.green()
        sound.play("do")
        dc_motor.go()
        sleep(0.5)
        dc_motor.stop()
    elif light_touch_sensor.is_pressed():  (условие можно изменить)
        # состояние неожиданности
        led.face("surprise")
        master.blue()
        sound.play("fx1")
    else:
        #сонное состояние
        led.face("sleepy")
        master.off()
    sleep(0.3)
```

Педагогу: дети самостоятельно выбирают эмоциональные состояния и реакции робота. При оценке учитывайте замысел карты эмоций, структуру if/elif/else и согласованную работу исполнительных блоков.

Занятие 9. Создание машины для викторин

✏ ШАГ 1 Анализ кода

questions — список текстов вопросов.
answers — список правильных ответов: верно или неверно.
Начальное значение score — 0. Правильный ответ увеличивает счёт на единицу.
touched == answers[i] сравнивает состояние кнопки с правильным ответом на текущий вопрос.

Пример функции подсчёта правильных ответов:

```
# Добавить в конце
rate = score / len(questions) * 100
print(f"Accuracy: {rate}%")
if rate >= 80:
    led.face("happy")
    master.green()
else:
    led.face("sad")
```

Примеры ответов для размышления (руководство, а не окончательные ответы)

list[index] обращается к элементу списка по индексу. Например, questions[0] — первый вопрос. Индексация начинается с нуля.

• Оценка = оценка + 1: добавьте 1 к текущему счету и сохраните снова. Базовая модель накопления.

Занятие 10. Итоговый проект по управлению роботом

✏ ШАГ 1 Ответы на тест по ключевым понятиям

Последовательное выполнение: код выполняется по одной строке, сверху вниз (занятие 1).
Цикл for: повторяет действия заданное число раз (занятия 1–2).
Список: Сохраняет несколько элементов данных по порядку в одной переменной (занятие 3).
Условный оператор: выбирает действия в зависимости от условия (занятие 5).
Функция (def): именованный фрагмент кода, который можно вызывать повторно (занятие 7).
Переменная: имя, с которым связывают значение в программе (все занятия).
Датчик получает сведения о состоянии среды или воздействии и преобразует их в данные программы.

Педагогу: у самостоятельного проекта нет единственного правильного решения. Совместное применение трёх и более функций — высокий уровень, двух — средний, одной — начальный. Оценивайте и ход работы над незавершённым проектом.

Занятие 11. Обучение модели Teachable Machine

ШАГ 3 Код Ответ:

```
from cubroid import dc_motor, master, sleep
from cubroid.ai import camera, teachable

camera.start()
teachable.load("https://teachablemachine.../model_url") ◂ (адрес своей обученной модели)
sleep(2)

while True:
    result = teachable.predict()
    name = result["label"]                ◂
    if name == "Class1Name":              ◂
        dc_motor.go()
        master.green()
    elif name == "Class2Name":            ◂
        dc_motor.back()
        master.red()
    sleep(0.5)
```

Проверено по API от 11.11.2026: teachable.load() и teachable.predict() доступны; результат содержит label и score. Нужны адрес собственной обученной модели и совпадающие имена классов.

Примеры ответов для размышления (руководство, а не окончательные ответы)

• Для обучения подготовьте достаточно разнообразных изображений. Например, начните с 30 снимков на класс и отдельно проверьте модель на новых кадрах.

• Смещение данных: если фотографировать только при ярком освещении, распознавание не удастся в темноте. Необходимо обучение в различных условиях.

Занятие 12. Распознавание цвета

Код Ответ:

```
from cubroid import dc_motor, master, led, sleep
from cubroid.ai import camera, color

camera.start()
sleep(1)

while True:
    c = color.name_en()
    if c == "red":              ◂ (name_en() возвращает английское имя цвета)
        master.red()
        dc_motor.go()           ◂
    elif c == "green":          ◂
        master.green()          ◂
        dc_motor.back()         ◂
    elif c == "blue":           ◂
        master.blue()
        dc_motor.stop()
    sleep(0.3)
```

Проверено по API от 11.11.2026: color.name_en() возвращает английское имя цвета, например red. Строки цветов в условии сохраняют на английском.

Занятие 13. Распознавание и синтез речи

✏ ШАГ 1 Изучите

Оператор in проверяет, содержится ли одна строка в другой. Например, "go" in text вернёт True, если text содержит go.

Русские слова должны совпадать с условиями программы. В текущей платформе распознавание русской речи требует исправления; перед занятием проверьте поддержку языка.

Код Ответ:

```
from cubroid import dc_motor, master, led, sleep
from cubroid.ai import voice, tts

voice.start()
tts.say("Ready!")
sleep(2)

while True:
    text = voice.text()
    if text:
        if "go" in text:
            dc_motor.go()
            tts.say("Движение вперёд!")
        elif "stop" in text:
            dc_motor.stop()
            tts.say("Стоп!")
        elif "left" in text:  (вариант учащегося)
            dc_motor.left()
            tts.say("Turning left!")
    sleep(0.2)
```

Совет преподавателю: команды и ответы TTS выбираются учащимися. 5+ отлично. Ключ: шаблон поиска строки с использованием оператора in.

Занятие 14. Совместная работа средств распознавания

✏ ШАГ 1 Блок-схема

Вход 1: Камера (распознавание цвета) → Процесс: определение цвета → Выход: двигатель/светодиод

Вход 2: Микрофон (распознавание речи) → Процесс: Поиск команды → Выход: Мотор/TTS

Если использованы независимые if, за один проход могут выполниться оба условия. В if/elif выполняется первая подходящая ветвь. Приоритет нужно задать явно.

Педагогу: оцените обоснованность сочетания функций, блок-схему и результаты проверки. Само увеличение числа функций не делает проект лучше.

Занятие 15. Распознавание рук

Код светодиодного табло Ответ:

```
# Добавить к существующему коду
if result == "win":        # если я выиграл
    score = score + 1
    led.number(score)
    master.green()
    sound.play("do")
```

✏ Теоретическая вероятность

При равновероятном независимом выборе соперника вероятность победы в «Камень, ножницы, бумага» — 1/3, около 33,3%.

random.choice() has uniform distribution, so it converges to 33% long-term.

В десяти раундах доля побед может заметно отличаться от 1/3. Отдельно обсудите влияние привычек игрока.

Занятие 16. Обнаружение лица и распознавание выражения

Код Ответ:

```
from cubroid import led, master, sound, sleep
from cubroid.ai import camera, face_mesh

camera.start()
face_mesh.start()
sleep(2)

while True:
    expr = face_mesh.expression()
    if expr:
        if expr == "smile":
            led.face("happy")
            master.green()
            sound.play("do")
        elif expr == "surprise":
            led.face("surprise")
            master.blue()
        elif expr == "eyes_closed":
            led.face("sleepy")
            master.off()
    sleep(0.5)
```

Проверено по API от 11.11.2026: face_mesh.expression() возвращает smile, big_smile, surprise, eyes_closed или neutral. Это рабочие значения: улыбка, широкая улыбка, удивление, закрытые глаза и нейтральное выражение.

✏ Руководство по этике ИИ

Для обсуждения: где полезно обнаружить наличие лица и где нужно установить личность человека? Учебный пример решает первую задачу.

Опасные случаи: слежка без согласия, расовая/гендерная предвзятость, личное слежение.

Единственного ответа нет. Попросите обосновать позицию примерами и учесть согласие участников.

Занятие 17. Распознавание позы

✏ **ШАГ 1. Понимание координат.**

Начало экранных координат — слева вверху. Координата y увеличивается вниз.
Обе руки подняты, если координаты y обоих запястий меньше координаты y носа.
(Если координата y запястья меньше координаты y носа = рука находится над носом)

Педагогу: в изображении y растёт вниз. Поэтому меньшая координата запястья означает положение выше носа. prev_action помогает не повторять одно действие непрерывно.

Занятие 18. Распознавание предметов и классификация звука

✏ ШАГ 1. Справочник по обнаружению объектов.

Ключевые объекты из набора данных COCO (80 классов):
человек, велосипед, автомобиль, стул, чашка, бутылка, сотовый телефон,
ноутбук, клавиатура, книга, ножницы, плюшевый мишка и т. д.
score показывает уверенность модели. Его шкалу и порог проверяют в используемом API; это не гарантия правильного ответа.

Код Ответ:

```
# Реакция на обнаружение объекта
if name == "person":
    dc_motor.go()          ◂
    master.green()         ◂
elif name == "cup":        ◂ (вариант учащегося)
    dc_motor.back()        ◂
    master.blue()          ◂
elif name == "book":       ◂ (вариант учащегося)
    dc_motor.stop()
    master.red()
```

Педагогу: дети самостоятельно обучают модель TM Audio различать не менее двух классов звуков, например хлопки и речь. Для более точного распознавания записывайте примеры в тихом помещении.

Занятие 19. Итоговый проект с ИИ: замысел и реализация

✏ **Нет фиксированного ответа – Руководство по оцениванию**

При проверке замысла:
✓ Ясно ли определение проблемы?
✓ Оправдан ли выбор технологии?
✓ Логична ли блок-схема?
✓ Правильно ли распределены роли?
✓ Реален ли пошаговый план?

Контрольные точки реализации:
✓ Соответствует ли программа замыслу?
✓ 50%+ реализовано?
✓ Проверена ли базовая работа?

Совет преподавателю: неполное задание — это нормально. Оценка на основе процесса. Сам процесс планирования → реализации → тестирования — это обучение. Не превышайте 15 минут планирования на занятии 19.

Занятие 20. Представление проекта и обсуждение результатов

✏ **Нет фиксированного ответа – Руководство по оцениванию**

Ключевые моменты оценки презентации:

✓ Есть ли структура: задача → решение → демонстрация → выводы?

✓ Понятно ли аудитории техническое объяснение?

✓ Стабильно ли демо? (Можно заменить объяснением процесса, если демонстрация не удалась)

✓ Аргументированно ли участники отвечают на вопросы?

Вопросы для обсуждения результатов:

✓ Описан ли опыт роста конкретно?

✓ Изменились ли взгляды на программирование/ИИ?

✓ Есть ли у ученика собственное мнение о компетенциях эпохи ИИ?

Педагогу: отведите группе 3–4 минуты. Заранее сохраните снимки работы на случай сбоя демонстрации. Просите подтверждать оценку конкретными примерами.

Конец ответов и пояснений для педагога

Команды сверены с исходниками Python Studio от 11.11.2026. Пропуски в рабочих листах заполняют учащиеся; адреса обученных моделей подставляет педагог. Перед занятием выполните пробу на своём наборе: проверка редакции не заменяет испытание оборудования.