

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ АМУРСКОЙ ОБЛАСТИ

ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ УЧРЕЖДЕНИЕ  
ДОПОЛНИТЕЛЬНОГО ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ  
«АМУРСКИЙ ОБЛАСТНОЙ ИНСТИТУТ РАЗВИТИЯ ОБРАЗОВАНИЯ»  
(ГАУ ДПО «АМИРО»)

ДЕТСКИЙ ТЕХНОПАРК «КВАНТОРИУМ-28»

**СБОРНИК УЧЕБНЫХ ЗАДАЧ  
ДЕТСКОГО ТЕХНОПАРКА «КВАНТОРИУМ-28»**

Благовещенск, 2024

ББК 74.04  
УДК 372.8  
С.23

Печатается по решению  
редакционно-издательского совета  
ГАУ ДПО «АМИРО»

Сборник учебных задач педагогов дополнительного образования Детского технопарка «Кванториум-28»: учебно-методическое пособие / Сост.: коллектив авторов – Благовещенск, Свободный 2024. – 102 с.

Общая редакция: Золотарёва Г.В., зам. директора ДТ «Кванториум-28», Рымина Е.Е., методист ДТ «Кванториум-28»

В сборник вошли задачи познавательного и занимательного характера, которые позволяют сделать процесс обучения интересным. Задачи сгруппированы по направлениям и по темам в соответствии с дополнительной образовательной общеразвивающей программой.

Сборник задач будет полезен в практической деятельности педагогам дополнительного образования Кванториумов и «Точек роста», а также всем, кто реализует программы «Робототехника», «Промышленный дизайн», «Виртуальная реальность», «Информационные технологии», «Компьютерное моделирование и др. в общеобразовательных организациях.

# ОГЛАВЛЕНИЕ

<b>ВВЕДЕНИЕ</b> .....	<b>Ошибка! Закладка не определена.</b>
<b>Раздел 1: «Робототехника»</b> .....	<b>8</b>
<b>LEGO Mindstorms Education EV3</b> .....	<b>8</b>
Задача № 1 «Механическая передача» .....	8
Задача № 2 «Над пропастью» .....	9
Задача № 3 «Калькулятор» .....	10
Задача № 4 «Температурный регулятор» .....	10
Задача № 5 «Инфракрасный лидар» .....	11
Задача № 6 «Тренажёр хирурга» .....	11
<b>Arduino</b> .....	<b>13</b>
Задача № 1 «Светодиод» .....	13
Задача № 2 «Светофор» .....	14
Задача № 3 «Индикатор температуры» .....	15
Задача № 4 «Потенциометр и сервопривод» .....	16
Задача № 5 «LCD индикатор и датчик температуры» .....	18
Задача № 6 «Светодиодная шкала» .....	19
<b>VEX IQ</b> .....	<b>20</b>
Задача № 1 «Робот с датчиком цвета» .....	20
Задача № 2 «Мигалка для робота» .....	21
Задача № 3 «Движение по линии» .....	22
Задача № 4 «Сторож» .....	22
Задача № 5 «Подъёмник» .....	22
<b>Раздел 2: «Компьютерное моделирование в среде автоматизированного проектирования Компас 3D»</b> .....	<b>24</b>
Задача № 1 «Трёхмерное построение многогранников в Компас 3D» .....	24
Задача № 2 «Построение правильной пирамиды» .....	26
Задача № 3 «Построение цилиндра операцией выдавливания» .....	28
Задача № 4 «Построение конуса операцией вращения» .....	30
Задача № 5 «Построение тора операцией вращения» .....	32
Задача № 6 «Трёхмерное моделирование сложных тел. Построение составной пирамиды» .....	33
Задача № 7 «Трёхмерное моделирование тела с применением кинематической операции «Элемент по траектории» .....	37

Задача № 8: «Трехмерное моделирование тела с применением кинематической операции «Элемент по сечениям»» .....	39
Задача № 9 «Пятиконечная звезда» .....	46
Задача № 10 «Автолиния, авторазмер и проекция» .....	47
Задача № 11 «3D-надпись» .....	48
Задача № 12 «Гофрированная труба» .....	49
Задача № 13 «Ребристый цилиндр» .....	49
Задание № 14 «Вентилятор» .....	51
Задание № 15 «Ваза» .....	52
Задание № 16 "Болт" .....	54
Задание № 17 «Новогодняя звезда» .....	55
Задание № 18 «Деталь сложной конфигурации» .....	56
<b>Раздел 3: «Информационные технологии» .....</b>	<b>56</b>
<b>Пиктопрограммирование и программирование в среде «Scratch3» .....</b>	<b>56</b>
Задача № 1 «Пиктопрограммирование» .....	56
Задача № 2 «Пиктопрограммирование» .....	57
Задача № 3 «Работа с пером в Scratch3» .....	58
Задача № 4 «Олимпийские кольца» .....	59
Задача № 5 «Таймер в Scratch3» .....	59
<b>Программирование в среде Unity .....</b>	<b>60</b>
Задача № 1 «Перемещение противника» .....	60
Задача № 2 «Переход между сценами» .....	65
Задача № 3 «Контроллер персонажа» .....	65
Задача № 4 «Двухмерный уровень» .....	66
Задача № 5 «Источник света» .....	67
Задача № 6 «Движущаяся платформа» .....	68
Задача № 7 «Параллакс эффект» .....	69
Задача № 8 «Звуковая дорожка» .....	70
<b>Программирование на языке Python и C# .....</b>	<b>72</b>
Задача № 1 «Работа с вводом/выводом» .....	72
Задача № 2 «Цикл for» .....	72
Задача № 3 «Условный оператор» .....	72
Задача № 4 «Количество гласных букв» .....	72
Задача № 5 «Проверка на простое число» .....	73
Задача № 6 «Перевод температуры» .....	73
Задача № 7 «Генерация случайных чисел» .....	73

Задача № 8 «Определение возрастной категории».....	74
Задача № 9 «Число и делители».....	74
Задача № 10 «Арифметические действия» .....	74
Задача № 11 «Простые числа» .....	75
Задача № 12 «Квадратное уравнение» .....	79
Задача № 14 «Среднее арифметическое».....	78
Задача № 16 «Дополни код».....	80
Задача № 17 «Калькулятор» .....	81
Задача № 18 «Отгадай число» .....	82
Задача № 19 «Больше, меньше или равно?» .....	83
Задача № 20 «Только нечётные» .....	84
Задача № 21 «Подсчёт суммы» .....	84
Задача № 22 «Поиск максимума» .....	85
Задача № 23 «Палиндром» .....	86
Задача № 24 «Факториал» .....	86
Задача № 25 «Сортировка списка».....	87
Задача № 26 «Подсчёт гласных» .....	87
<b>Программирование на языке C++.....</b>	<b>88</b>
Задача № 1 «Работа с вводом/выводом».....	88
Задача № 2 «Работа со стандартными математическими функциями» .....	88
Задача № 3 «Работа с операторами сравнения» .....	89
Задача № 4 «Условные операторы» .....	89
Задача № 5 «Работа с циклом for».....	89
Задача № 6 «Работа с пользовательским вводом».....	89
Задача № 7 «Работа с библиотекой математических функций» .....	90
Задача № 8 «Факториал в C++».....	90
<b>Раздел 4: «Виртуальная реальность и промышленный дизайн» .....</b>	<b>91</b>
<b>Unity.....</b>	<b>91</b>
Задача № 1 «Расстановка по пьедесталам».....	91
Задача № 2 «Мишень» .....	92
Задача № 3 «Виртуальный музей» .....	92
Задача № 4 «Приложение для медитации».....	93
Задача № 5 «Приложение для тренировки реакции» .....	94
Задача № 6 «Приложение для тренировки меткости».....	95
<b>Blender.....</b>	<b>98</b>
Задача № 1 «Цыплёнок» .....	98

Задача № 2 «Ромашка» .....	98
Задача № 3 «Мебель».....	99
Задача № 4 «Тарелка».....	100
Задача № 5 «Башня» .....	98
Задача № 6 «Шахматная доска» .....	100

## ВВЕДЕНИЕ

Целью деятельности детского технопарка «Кванториум-28» является создание и развитие системы современных инновационных площадок интеллектуального развития и досуга для детей и подростков на территории России.

Детские технопарки – это площадки дополнительного образования, оснащённые высокотехнологичным оборудованием, нацеленные на подготовку новых высококвалифицированных инженерных кадров, разработку, тестирование и внедрение инновационных технологий и идей.

Здесь школьники получают основы инженерного образования, навыки проектной деятельности и решают реальные производственные задачи в сопровождении опытных наставников, в том числе представителей научной школы, промышленности и бизнеса. Занятия проводятся с использованием технологии ТРИЗ (теория решения изобретательских задач), которая успешно развивает 4К-компетенции (коммуникация, креативность, командное решение проектных задач, критическое мышление) обучающихся.

Детский технопарк «Кванториум-28» функционирует в рамках регионального проекта, обеспечивающего достижение целей, показателей и результатов федерального проекта «Успех каждого ребенка» национального проекта «Образование».

В данном сборнике представлены учебные практические задачи педагогов дополнительного образования детского технопарка «Кванториум-28», площадки детского технопарка «Кванториум-28» г. Свободного и мобильного детского технопарка «Кванториум-28».

## РАЗДЕЛ 1: «РОБОТОТЕХНИКА»

Аргудаева Анна Андреевна,  
педагог дополнительного образования  
ДТ «Кванториум-28»

### LEGO Mindstoms Education EV3

**Возрастная категория:** задачи тематики LEGO Mindstoms Education EV3 предназначены для обучающихся возрастом от 9 до 11 лет, закончивших обучение вводного модуля в качестве закрепления материала или начальных задач для обучающихся углубленного модуля.

#### Задача № 1 «Механическая передача»

Механическая передача – это механизм для преобразования (передачи) вращательного движения с ведущего вала различных машин на ведомый.

При помощи механической передачи в приводах осуществляется:

- повышение или понижение скорости;
- её ступенчатое либо бесступенчатое регулирование;
- изменение направления движения;
- приведение в движение нескольких механизмов одним двигателем.

Механические передачи бывают двух типов:

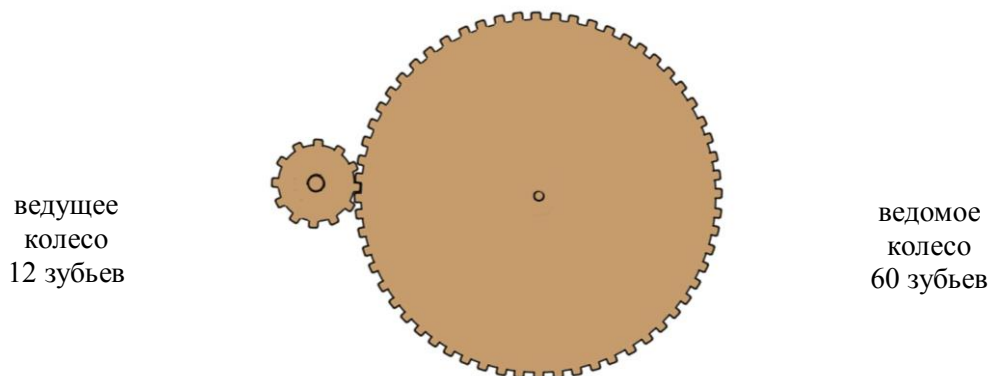
- с непосредственным касанием закреплённых на валах деталей;
- с включением промежуточной гибкой связи (ремень, канат и др.).

Одной из разновидностей механической передачи с непосредственным касанием закреплённых на валах деталей – зубчатая передача.

Передаточное число – соотношение количества зубьев двух взаимодействующих шестерён, ведомой к ведущей.

Для расчёта передаточного числа используется формула:

$$N1/N2 \text{ например } 12/60 = 1/5$$



**Условие:** сконструируйте зубчатую передачу со следующим передаточным числом:

1.  $1/3$
2.  $1/1$  используя нечетное количество шестеренок
3.  $25/1$  используя двухступенчатую передачу

**Возможное решение:**

1. Построить одноступенчатую передачу из двух шестерней: 8 зубцов и 24 зубца.
2. Построить одноступенчатую передачу из пяти шестерней:  $24 + 8 + 40 + 8 + 24$ .
3. Строим двухступенчатую передачу, используя одну прямую балку:



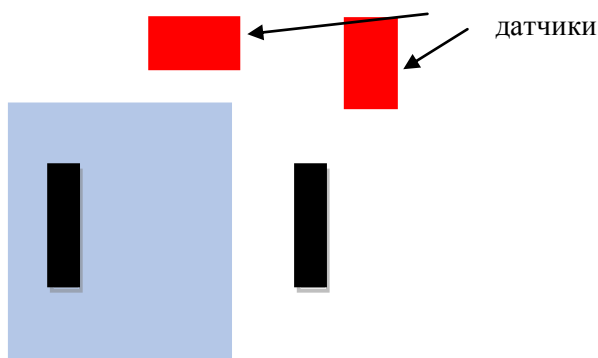
С одной стороны балки используем в качестве ведущей шестеренки 40-зубчатую, в качестве ведомой – 8-зубчатую. С другой стороны балки на ту же ось, на которой прикреплена 8-зубчатая шестеренка, присоединяем 40-зубчатую – она будет ведущей. Ведомой шестеренкой присоединяем 8зубую шестеренку.

## Задача № 2 «Над пропастью»

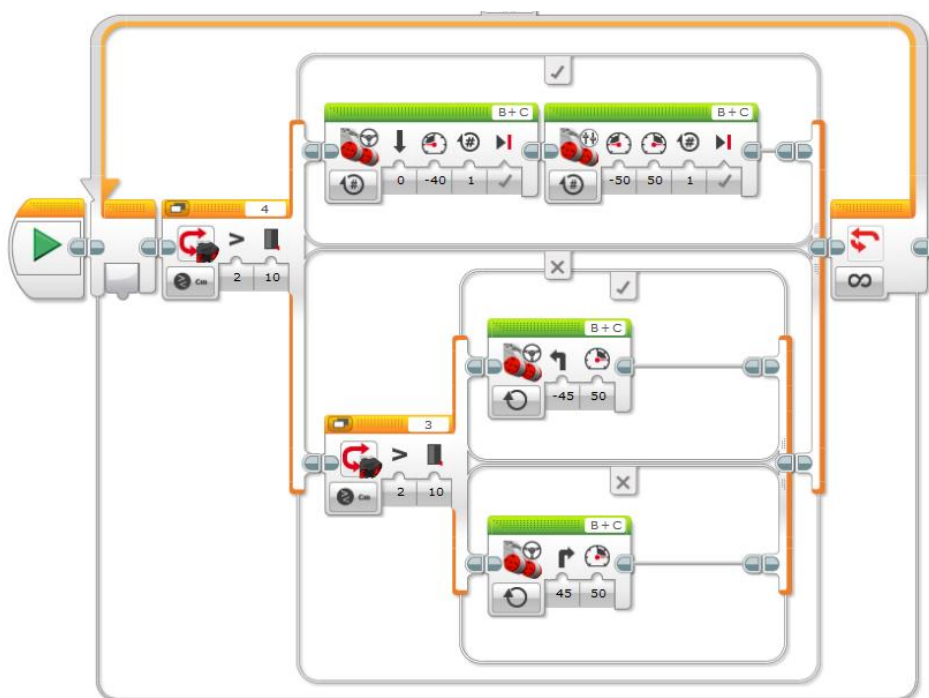
**Условие:** сконструируйте и запрограммируйте колёсного робота, способного проезжать вдоль «пропасти». В качестве «пропасти» может выступать край парты. «Пропасть» находится с одной стороны робота (справа или слева). Также робот должен предотвратить съезд в «пропасть» вперед.

### Возможное решение:

Любая колёсная модель колёсного робота. Впереди по правой стороне робота установить ультразвуковой или инфракрасный датчик, направленный ровно вниз (его проекция должна выступать за проекцию колеса как вперед, так и вправо). Второй ультразвуковой или инфракрасный датчик необходимо установить впереди по центру робота, проекция датчика располагается перед колесами, датчик направлен ровно вниз.



### Программа:



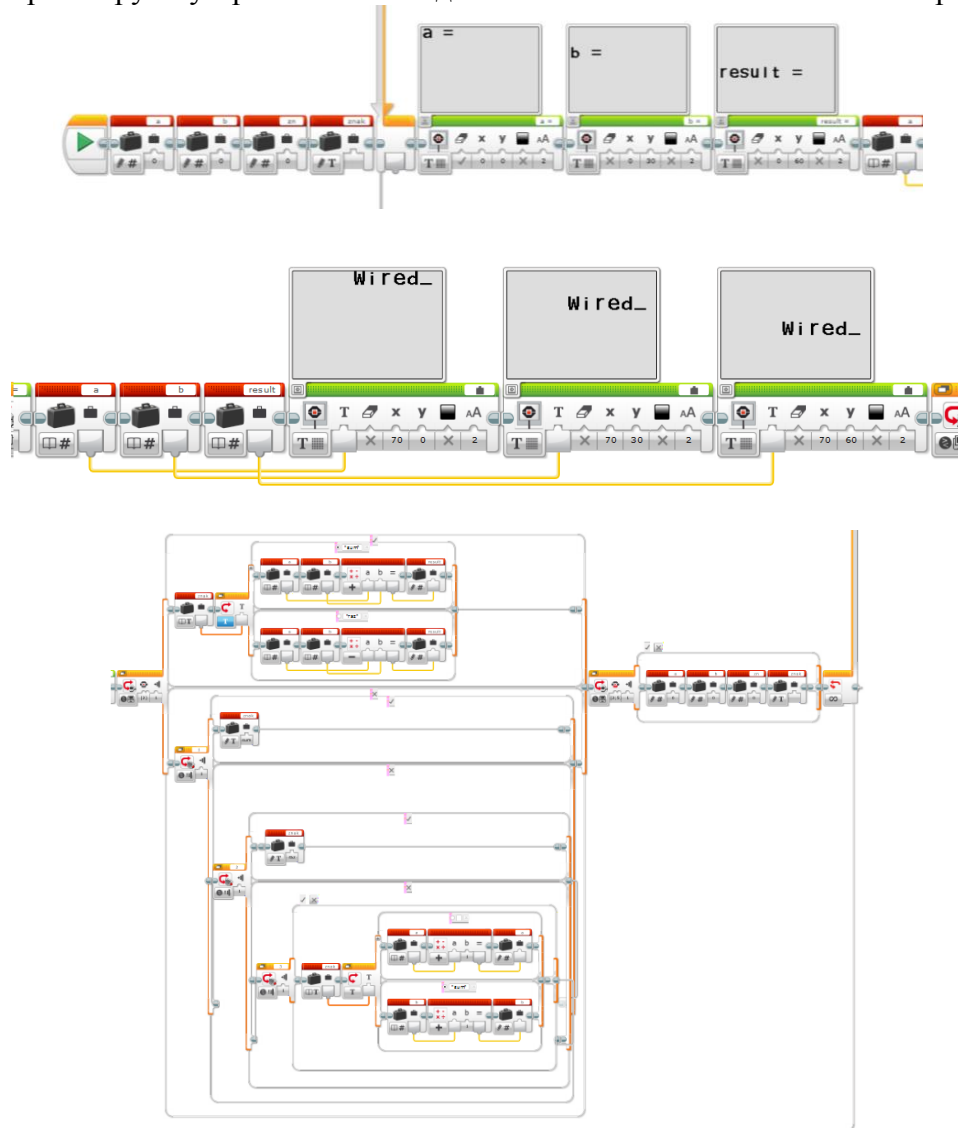
### Задача № 3 «Калькулятор»

**Условие:** сконструируйте устройства с тремя датчиками касания (кнопками). Запрограммируйте его на выполнение операции сложение и вычитание цифр.

**Возможное решение:**

1. Конструируется модель устройства, состоящая из микроконтроллера, двух датчиков касания правой стороны и 1 датчика касания с левой стороны микроконтроллера. Датчик с левой стороны будет выполнять функцию выбора ввода цифры. Два датчика с правой стороны отвечают за математические операции.

2. Програмируем устройство на ввод и выполнение математических операций.



### Задача № 4 «Температурный регулятор»

**Условие:** сконструируйте и запрограммируйте устройство, которое будет включать вентилятор при превышении температуры помещения выше 22 градусов.

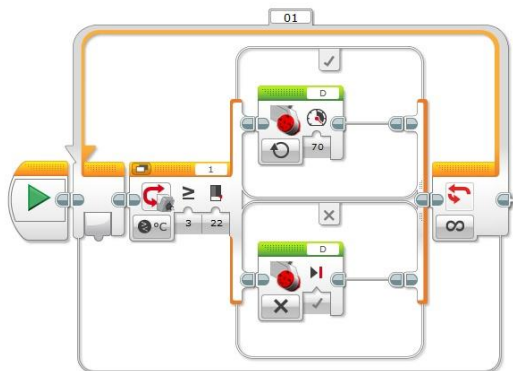
**Возможное решение:**

1. Конструкция представляет из себя настольный вентилятор с установленным сбоку датчиком температуры.

2. В программном блоке «Цикл» располагается блок «Переключатель» в режиме «Датчик температуры» - «Сравнение» - «По Цельсию», тип сравнения «больше или равно» с пороговым значением равным 22.

3. При выполнении условия программный блок «Большой мотор» в режиме «Включить» со значением скорости 70.

4. При невыполнении условия программный блок «Большой мотор» в режиме «Выключить».



### Задача № 5 «Инфракрасный лидар»

**Условие:** сконструируйте и запрограммируйте робота, вращающегося вокруг своей оси и останавливающегося в направлении другого инфракрасного датчика.

**Возможное решение:**

Колесная модель робота с расположенным впереди и направленным вперед ИК датчиком.

**Программа:**

1. Используя программный блок «Независимое управление моторами», начать вращение робота вокруг своей оси против часовой стрелки.

2. Используя программный блок «Ожидание» в режиме «Инфракрасный датчик» - «Сравнение» - «Приближение маяка» с пороговым значением равным 80, ожидаем, пока робот не обнаружит инфракрасный маяк (значение параметра «Приближение» станет меньше 100).

3. Так как наш робот вращается против часовой стрелки, то, когда инфракрасный датчик обнаружит маяк, его параметр «Направление» примет отрицательное значение. Поэтому, следующий программный блок «Ожидание» в режиме «Инфракрасный датчик» - «Сравнение» - «Направление маяка» даст возможность роботу вращаться до тех пор, пока робот не окажется напротив инфракрасного маяка (значение параметра «Пороговое значение» превысит 0).

4. Так как наш робот, вращаясь с большой скоростью, может повернуть чуть больше в результате сил инерции, то, на малой скорости, используя следующие два программных блока, повернем робота по часовой стрелке.

5. Выключим моторы робота.



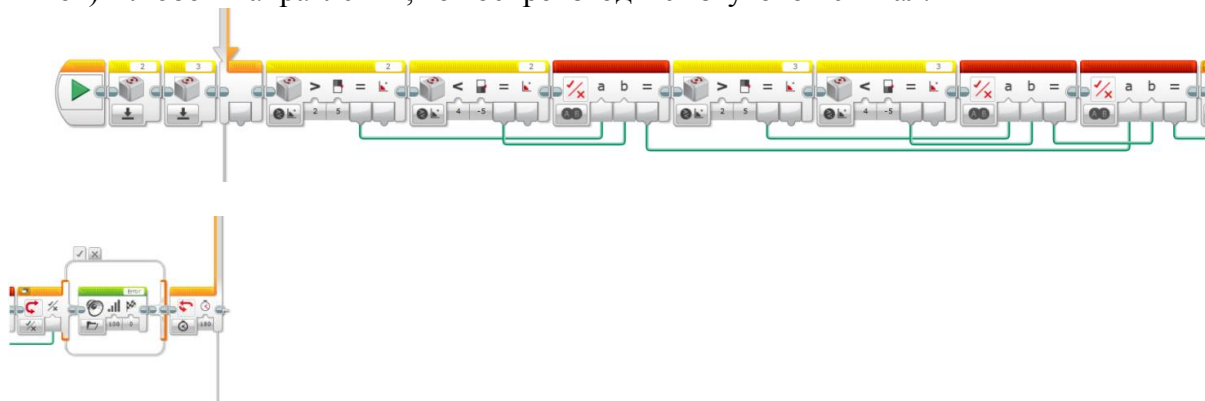
### Задача № 6 «Тренажёр хирурга»

**Условие:** сконструировать и запрограммировать модель, которая будет выполнять функцию тренажера для будущего хирурга. Модель должна представлять из себя устройство, фиксирующееся на кисти, запястье и предплечье руки человека (не гнущееся). На данном устройстве должны располагаться два гироскопических датчика, измеряющих углы наклона в разных плоскостях.

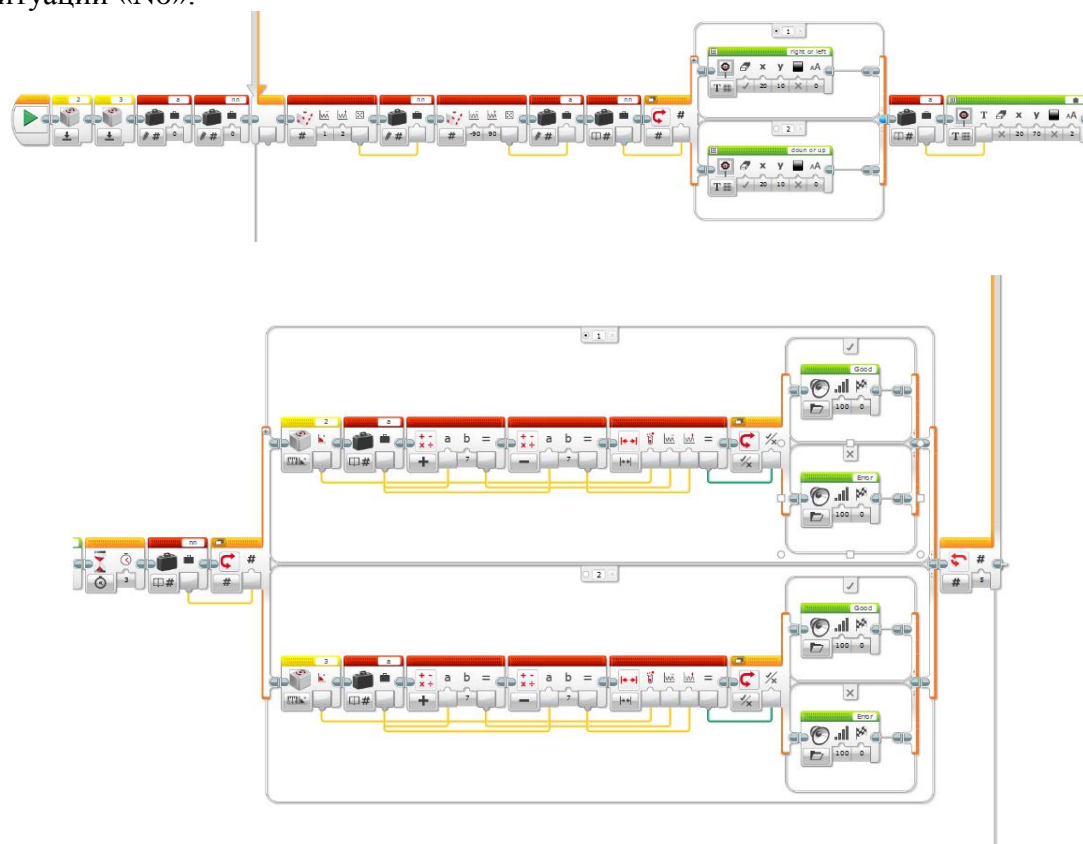
### Решение:

Запрограммируйте его на две функции:

1. Устройство должно измерить устойчивость положения руки человека в течение 3 минут. Если датчики фиксируют отклонения более чем на 5 градусов (погрешность датчиков) в любом направлении, то воспроизводится звуковой сигнал.



2. Устройство должно задать поочередно 5 случайных значений, на которое необходимо отклонить руку (например, вниз 45 градусов), если человек правильно наклоняет руку (с погрешностью в 5-10 градусов), воспроизводится звуковой сигнал «Yes», в другой ситуации «No».



## Arduino

### Задача № 1 «Светодиод»

**Условие:** подключите светодиод к плате Arduino и запрограммируйте его для мигания определённым образом.

**Материалы и оборудование:** плата Arduino любой модели, плата для прототипирования, резистор 220 Ом, соединительные провода, светодиод любого цвета.

#### Ход выполнения:

1. Подключите анод (длинный вывод) светодиода к цифровому выходу платы Arduino (например, к выходу 13).
2. Подключите катод (короткий вывод) светодиода к земле (GND) через сопротивление в 220 Ом для защиты Arduino от перегрузки.
3. Откройте среду разработки Arduino и создайте новый скетч.
4. В функции setup() пропишите задание выхода 13 (или другого выхода, к которому подключён светодиод) как выход.
5. В функции loop() пропишите цикл мигания светодиода с произвольной задержкой.
6. Загрузите скетч в плату Arduino и наблюдайте за миганием светодиода.

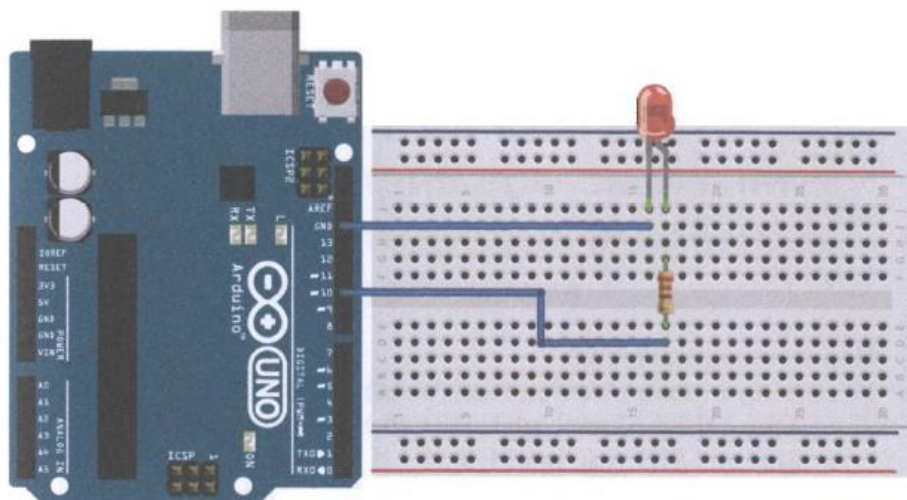
Опционально: вы можете изменить задержки в цикле мигания, чтобы изменить скорость мигания светодиода.

#### Приложение

Примерный листинг программы

```
pinMode(13, OUTPUT);  
void loop() {  
  digitalWrite(13, HIGH); // зажигаем светодиод  
  delay(1000);           // ждем 1 секунду  
  digitalWrite(13, LOW);  // гасим светодиод  
  delay(1000);           // ждем 1 секунду  
}
```

Пример схемы



## Задача № 2 «Светофор»

**Условие:** подключите светодиоды к плате Arduino и запрограммируйте его таким образом, чтобы схема работала как светофор.

**Материалы и оборудование:** плата Arduino любой модели, плата для прототипирования, резистор 220 Ом (3 шт.), соединительные провода, светодиоды (3 шт.) разных цветов.

### Ход выполнения:

1. Подключите светодиоды анодом (длинный вывод) к цифровому выходу платы Arduino (например, к выходам 4,7,10).
2. Подключите светодиоды катодом (короткий вывод) к земле (GND) через сопротивление в 220 Ом для защиты Arduino от перегрузки.
3. Откройте среду разработки Arduino и создайте новый скетч.
4. В функции setup() пропишите задание выходов 4,7,10 (или другие выходы, к которым подключены светодиоды) как выходы.
5. В функции loop() пропишите код, который будет заставлять три светодиода переключаться примерно с той задержкой, которая принята в реальных светофорах (от красного к зеленому).
6. Загрузите скетч в плату Arduino и наблюдайте за миганием светодиодов.

Опционально: вы можете изменить задержки в цикле мигания, чтобы изменить скорость мигания светодиодов, либо улучшить схему, добавив тактовую кнопку для переключения светодиодов по нажатию кнопки.

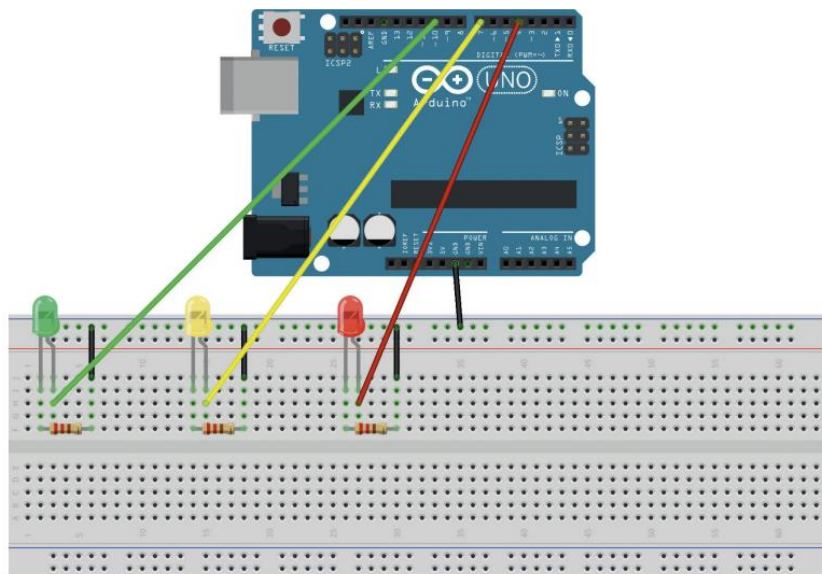
### Приложение

Примерный листинг программы

```
int redled = 10;
int yellowled = 7;
int greenled = 4;
void setup ()
{
  pinMode (redled, OUTPUT);
  pinMode (yellowled, OUTPUT);
  pinMode (greenled, OUTPUT);
}
void loop ()
{
  digitalWrite (redled, HIGH);
  delay (1000);
  digitalWrite (redled, LOW);
  digitalWrite (yellowled, HIGH);
  delay (200);
  digitalWrite (yellowled, LOW);
  digitalWrite (greenled, HIGH);
  delay (1000);
  digitalWrite (greenled, LOW);
}
```



## Пример схемы



### Задача № 3 «Индикатор температуры»

**Условие:** подключите датчик температуры и RGB светодиод к плате Arduino и запрограммируйте его таким образом, чтобы получить индикатор температуры окружающей среды.

**Материалы и оборудование:** Плата Arduino любой модели, плата для прототипирования, датчик LM335 (либо аналог), резисторы 220 Ом (3 шт.), резистор 2,2 кОм, соединительные провода, RGB-светодиод.

#### Ход выполнения:

1. Подключите датчик температуры LM35 к плате Arduino, используя аналоговый вход A0. Подключите вывод датчика к плюсу (+) 5-вольтовой линии питания, а ноль (-) к одному из аналоговых входов платы Arduino.
2. Подключите RGB светодиод к плате Arduino, используя три цифровых вывода для каждого цвета. Например, подключите красный вывод к цифровому выводу 9, зелёный вывод к цифровому выводу 10, и синий вывод к цифровому выводу 11.
3. Откройте среду разработки Arduino и создайте новый скетч.
4. В начале скетча объявите переменные для хранения значений температуры и цветов светодиода.
5. В функции `setup()` инициализируйте цифровые выводы для светодиода как выходы.
6. В функции `loop()` считайте значение температуры с датчика LM35 и преобразуйте его в градусы Цельсия.
7. Определите диапазон температур для каждого цвета светодиода. Например, от 0 до 20 градусов Цельсия - красный цвет, от 21 до 30 градусов Цельсия - оранжевый цвет, и т.д.
8. Установите значения цветов светодиода с помощью функции `analogWrite()` для каждого цвета.
9. Добавьте задержку в 1 секунду, чтобы обновлять значения температуры и цветов светодиода.
10. Сохраните скетч и загрузите его на плату Arduino, вы должны увидеть индикатор температуры окружающей среды, который отображается с помощью изменения цвета RGB светодиода.

#### Приложение

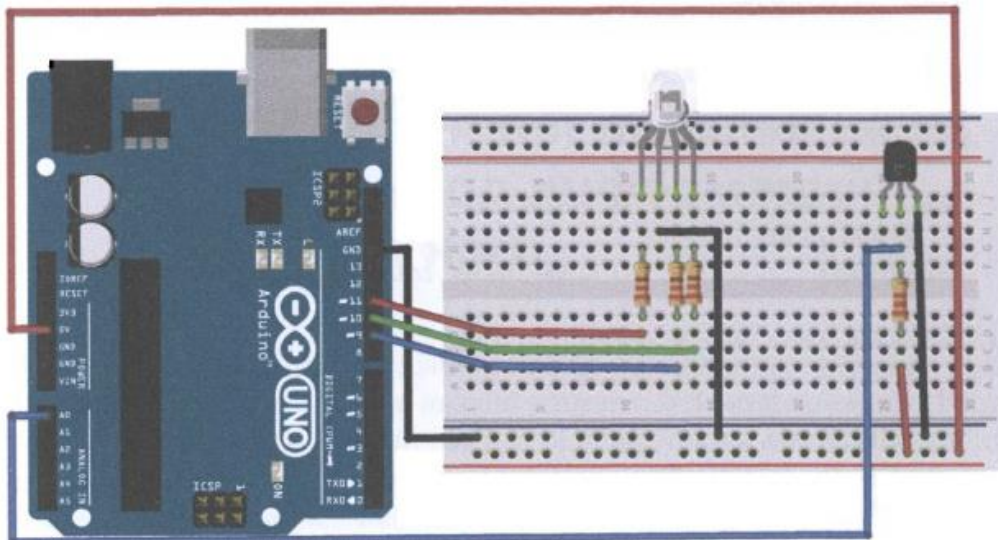
#### Примерный листинг программы

```

int temp = 0;
int red = 0;
int green = 0;
int blue = 0;
void setup() {
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  pinMode(11, OUTPUT);
}
void loop ()
{
  temp = analogRead(A0);
  temp = temp * 500 / 1024;
  if (temp >= 0 && temp <= 20) {
    red = 255;
    green = 0;
    blue = 0;
  }
  else if (temp > 20 && temp <= 30) {
    red = 255;
    green = 128;
    blue = 0;
  }
  analogWrite(9, red);
  analogWrite(10, green);
  analogWrite(11, blue);
  delay(1000);}

```

Пример схемы



#### Задача № 4 «Потенциометр и сервопривод»

**Условие:** подключите потенциометр и сервопривод к плате Arduino и запрограммируйте таким образом, чтобы при кручении потенциометра изменялся угол поворота сервопривода.

**Материалы и оборудование:** Плата Arduino любой модели, плата для прототипирования, сервопривод SG90 (либо аналог), потенциометр 1 кОм, соединительные провода.



### Ход выполнения:

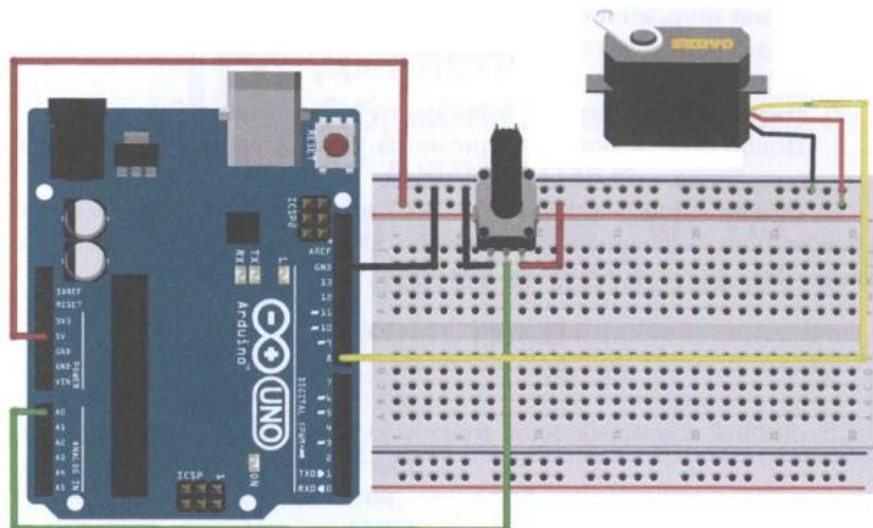
1. Подключите потенциометр к плате Arduino следующим образом:  
Первый вывод потенциометра (обычно обозначен как "VCC" или "+") подключите к 5-ти вольтовому выводу платы Arduino.  
Второй вывод потенциометра (обычно обозначен как "SIG" или "OUT") подключите к аналоговому входу платы Arduino (например, A0).  
Третий вывод потенциометра (обычно обозначен как "GND" или "-") подключите к земле платы Arduino.
2. Подключите сервопривод к плате Arduino следующим образом:  
Первый вывод сервопривода подключите к цифровому выводу платы Arduino (например, D9).  
Второй вывод сервопривода подключите к земле платы Arduino.  
Третий вывод сервопривода подключите к питанию платы Arduino (например, 5V).
3. В функции setup() инициализируйте аналоговый вход для потенциометра и цифровой вывод для сервопривода.
4. В функции loop() считывайте значение на аналоговом входе, которое изменяется в зависимости от положения регулировочного винта потенциометра. Преобразуйте значение с аналогового входа в значение угла поворота сервопривода (например, от 0 до 180 градусов). Установите угол поворота сервопривода с помощью функции servo.write(angle). Ждите некоторое время (например, 10 миллисекунд) с помощью функции delay(10).

### Приложение

#### Примерный листинг программы

```
#include <Servo.h>
Servo servo;
int servoPin = 9;
int potentiometerPin = A0;
void setup() {
  servo.attach(servoPin);
}
void loop() {
  int potentiometerValue = analogRead(potentiometerPin);
  int servoAngle = map(potentiometerValue, 0, 1023, 0, 180);
  servo.write(servoAngle);
  delay(10);
}
```

#### Пример схемы



## Задача № 5 «LCD индикатор и датчик температуры»

**Условие:** подключите LCD индикатор и датчик температуры к плате Arduino и запрограммируйте для вывода температуры на индикатор.

**Материалы и оборудование:** Плата Arduino любой модели, плата для прототипирования, резистор 2,2 кОм, резистор 51 Ом, LCD-экран WH1602 (или аналог), датчик температуры LM335 (либо аналог), соединительные провода.

### Ход выполнения:

1. Подключите датчик температуры DS18B20 к плате Arduino Uno следующим образом:

Вставьте датчик в соответствующий разъем на плате. Подключите вывод данных датчика (DQ) к цифровому выводу платы (например, D4). Подключите наземный провод датчика к наземному выводу платы (GND).

Подключите питающий провод датчика к питающему выводу платы (+5V).

2. Подключите 16x2 LCD индикатор к плате Arduino Uno следующим образом:

Подключите выводы RS, EN, D4, D5, D6, D7 индикатора к соответствующим цифровым выводам платы (например, RS к 2, EN к 3, D4 к 4, D5 к 5, D6 к 6, D7 к 7). Подключите наземный провод индикатора к наземному выводу платы (GND). Подключите питающий провод индикатора к питающему выводу платы (+5V).

3. Напишите код для платы Arduino Uno

Проверьте работу программы: после загрузки кода на плату Arduino Uno, текущая температура должна отображаться на 16x2 LCD индикаторе.

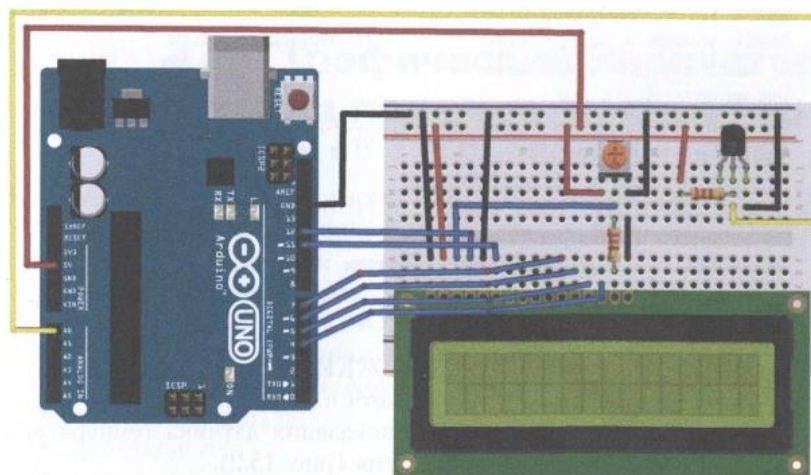
Примечание: если вы используете другие выводы для подключения датчика температуры и LCD индикатора, измените соответствующие значения в коде.

### Приложение

#### Примерный листинг программы

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(12, 11, 7, 6, 5, 4);
const int LM335=A0;
void setup() {
  lcd.begin(16, 2);
}
void loop()
{
  double val = analogRead(LM335);
  double voltage = val*5.0/1024;
  lcd.setCursor(2,0);
  lcd.print("Tk="); lcd.print(voltage*100); lcd.print("K");
  double temp = voltage*100 - 273.15;
  lcd.setCursor(2,1);
  lcd.print("Tc="); lcd.print(temp); lcd.print("");
  delay(1000);
}
```

#### Пример схемы



## Задача № 6 «Светодиодная шкала»

**Условие:** подключите 10-ти сегментную светодиодную шкалу и потенциометр к плате Arduino и запрограммируйте таким образом, чтобы при повороте ручки потенциометра изменялось количество светящихся светодиодов на 10-ти сегментной светодиодной шкалы.

**Материалы и оборудование:** плата Arduino любой модели, плата для прототипирования, потенциометр 2 кОм, резистор 220 Ом (10 шт.), светодиодная шкала 10 сегментов, соединительные провода.

### Ход выполнения:

1. Подключите 10-сегментную светодиодную шкалу к цифровым выводам платы Arduino. Например, вы можете подключить сегменты A-G к выводам 2-8, а десятый сегмент к выводу 9.
2. Подключите потенциометр к аналоговому вводу платы Arduino. Например, вы можете подключить ручку потенциометра к выводу A0.
3. Напишите программу на языке Arduino, которая будет считывать значение потенциометра с помощью функции `analogRead()` и сохранять его в переменную.
4. Определите максимальное и минимальные значения потенциометра, соответствующие минимальному и максимальному количеству зажигающихся светодиодов на светодиодной шкале. Например, если вы хотите, чтобы при минимальном значении потенциометра загорался только один светодиод, а при максимальном все 10, то минимальное значение будет 0, а максимальное - 1023 (максимальное значение для аналогового ввода Arduino).
5. Вычислите количество зажигающихся светодиодов на основе текущего значения потенциометра. Например, если текущее значение потенциометра равно 512, то зажигается половина светодиодов.
6. Выведите текущее количество зажигающихся светодиодов на светодиодную шкалу. Для этого используйте функции `digitalWrite()` и `delay()` для зажигания нужных сегментов светодиодной шкалы на определенное время.
7. Проверьте работу программы и измените количество зажигающихся светодиодов на светодиодной шкале в зависимости от поворота ручки потенциометра.

### Приложение

#### Примерный листинг программы

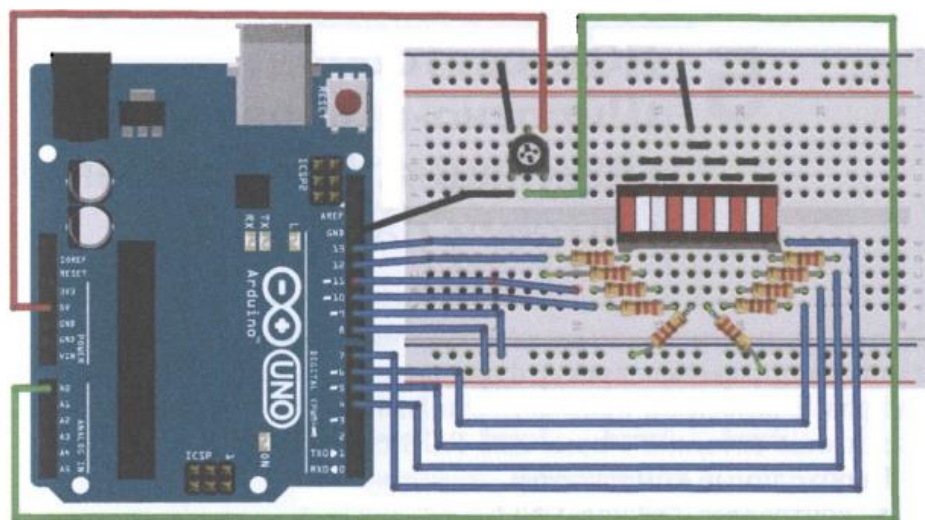
```
const int POT=0;
int valpot = 0;
const int pinsled[10]={3,4,5,6,7,8,9,10,11,12};
```

```

int countleds = 0;
void setup()
{
  for(int i=0;i<10;i++)
  {
    pinMode(pinsled[i],OUTPUT);
    digitalWrite(pinsled[i],LOW);
  }
}
void loop()
{
  valpot = analogRead(POT);
  countled=map(valpot,0,1023,0,10);
  for(int i=0;i<10;i++)
  {
    if(i<countleds)
      digitalWrite(pinsled[i],HIGH);
    else
      digitalWrite(pinsled[i],LOW);
  }
}

```

Пример схемы



*Дивиченко Иван Иванович,  
педагог дополнительного образования  
ДТ «Кванториум-28»*

## VEX IQ

### Задача № 1 «Робот с датчиком цвета»

**Условие:** напишите программу для робота, чтобы при использовании датчика цвета определялись цвета «Синий», «Зеленый», «Красный» и результат выводился на экране микроконтроллера, а также этим цветом горел датчик касания.

**Решение:**



## Задача № 2 «Мигалка для робота»

**Условие:** напишите программу «Мигалка» для робота с двумя установленными датчиками касания. Установите на первый датчик цвет «Красный», на второй цвет «Синий», интервал смены цветов не более 0,5 секунд, во время мигания также должен проигрываться звук «Сирена2» и количество повторов не менее 10, по завершению оба датчика должны быть выключены.

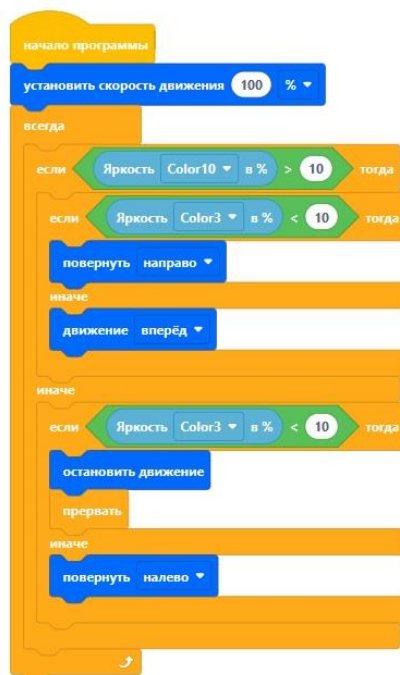
**Решение:**



### Задача № 3 «Движение по линии»

**Условие:** напишите программу для робота с двумя датчиками цвета, для езды по черной линии до перекрестка.

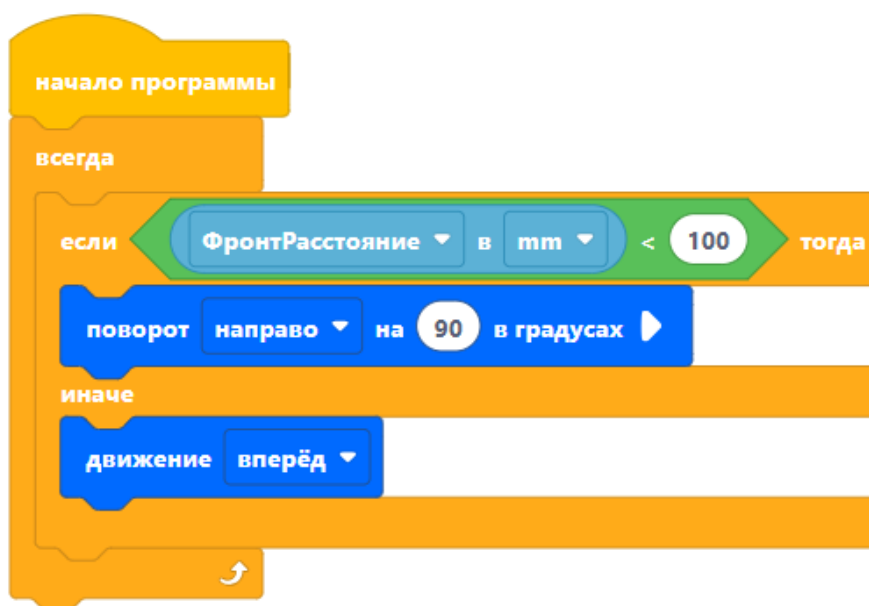
**Решение:**



### Задача № 4 «Сторож»

**Условие:** напишите программу с использованием ультразвукового датчика для робота «Сторож», при выполнении которой он должен поворачивать под прямым углом направо всегда, когда до стены расстояние будет равно 10 см.

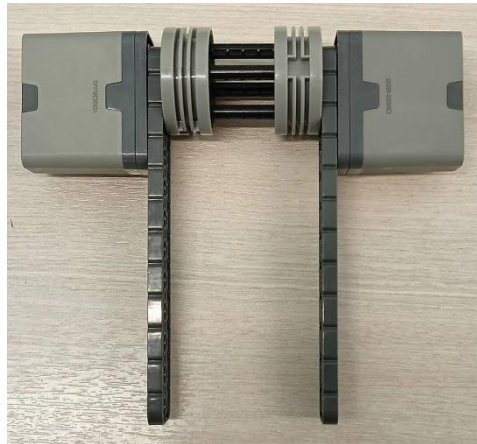
**Решение:**





## 5 «Подъёмник»

**Условие:** напишите программу для робота «Подъёмник», используя механизм как на картинке или подобный ему, для подъема по канату длиной 150 см.



«Подъемник» должен быть оснащен ультразвуковым датчиком, установленным в самой верхней точке робота и быть направленным вверх. Канат, протянутый через механизм, должен накручиваться на него. При подъеме наверх «Подъёмник» должен остановиться на расстоянии 5 см до преграды на 5 секунд, подать звуковой сигнал о спуске, спуститься вниз и подать сигнал о завершении спуска.

**Решение:**



## РАЗДЕЛ 2: «КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ В СРЕДЕ АВТОМАТИЗИРОВАННОГО ПРОЕКТИРОВАНИЯ КОМПАС 3D»

Пименов Тимофей Сергеевич,  
педагог дополнительного образования  
ДТ «Кванториум-28»

### Задача № 1 «Трёхмерное построение многогранников в Компас 3D»

**Возраст:** 10-13 лет

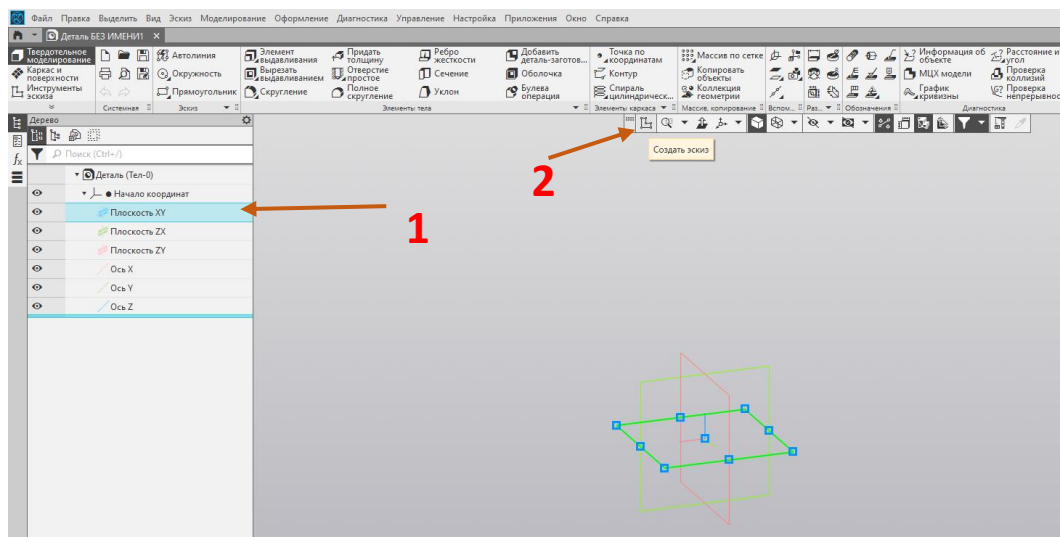
**Условие:** построить трёхмерную модель параллелепипеда в программе Компас 3D.

**Определения:** *Прямоугольный параллелепипед* – параллелепипед, все грани которого являются прямоугольниками.

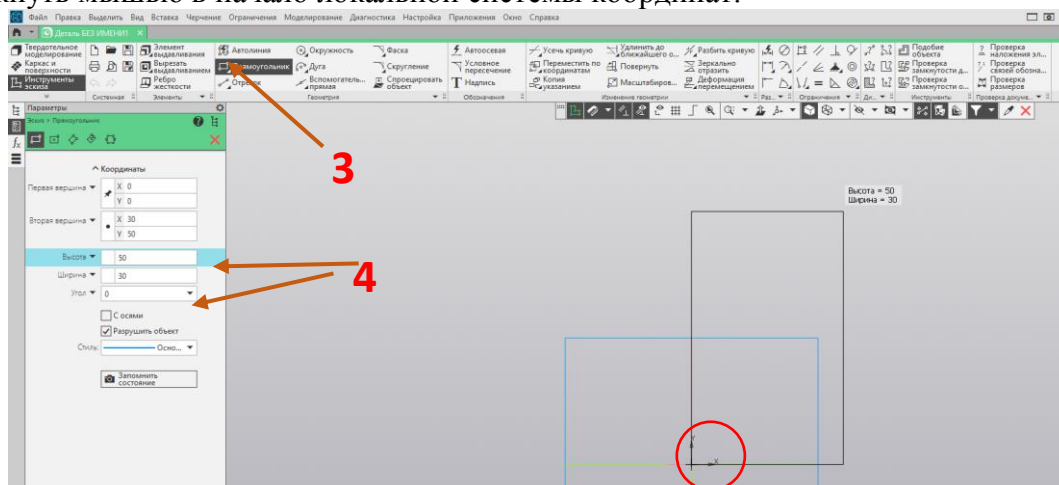
**Элемент выдавливания** – позволяет создать основание детали, представляющее собой тело выдавливания.

**Порядок выполнения:**

1. Запустить программу Компас 3D.
2. Выбрать создание детали (Файл-Создать-Деталь).
3. Выбрать в дереве модели плоскость XY.
4. Включить режим эскиз (кнопка панели управления).

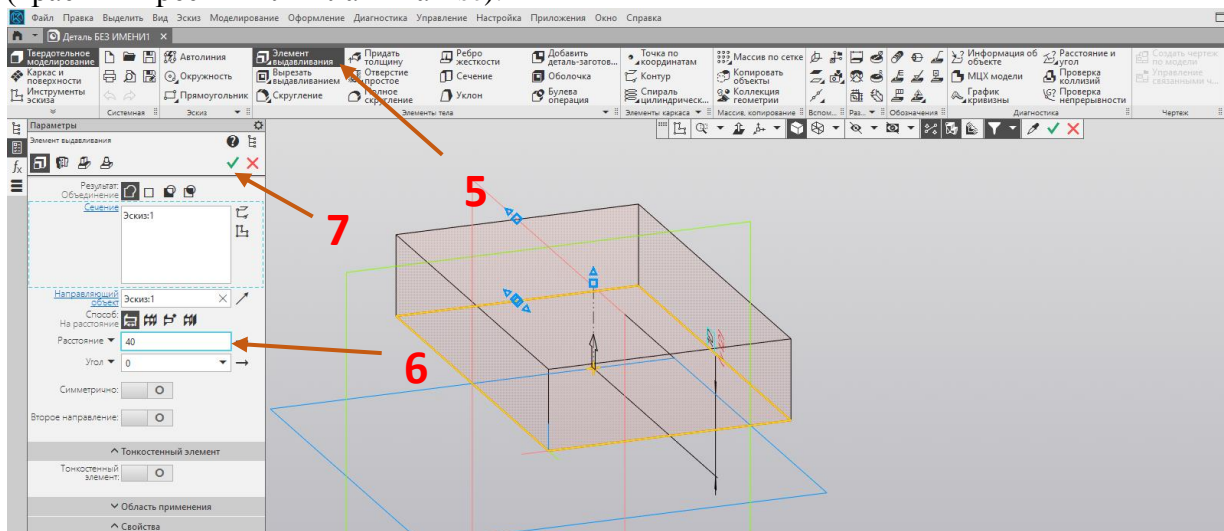


5. На панели **Геометрия** выбрать инструмент **Прямоугольник**.
6. В окне параметров ввести размеры: высота - 50; ширина - 30. Нажать ввод (Enter) и кликнуть мышью в начало локальной системы координат.

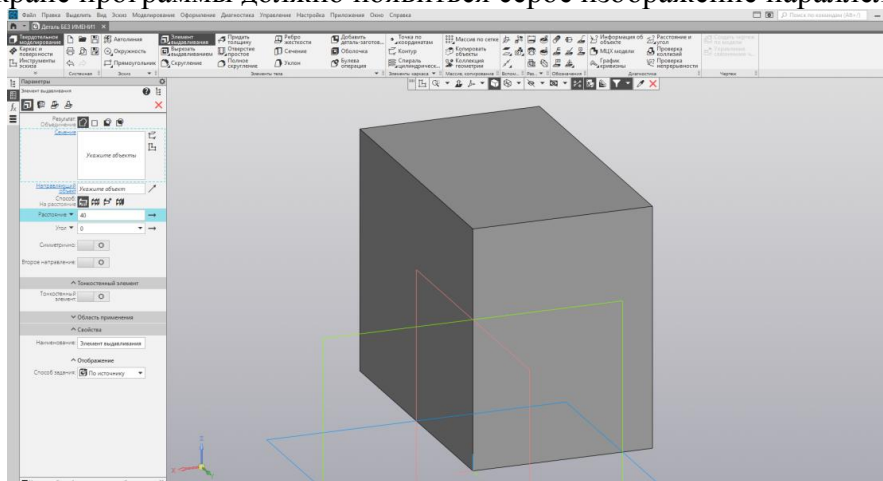




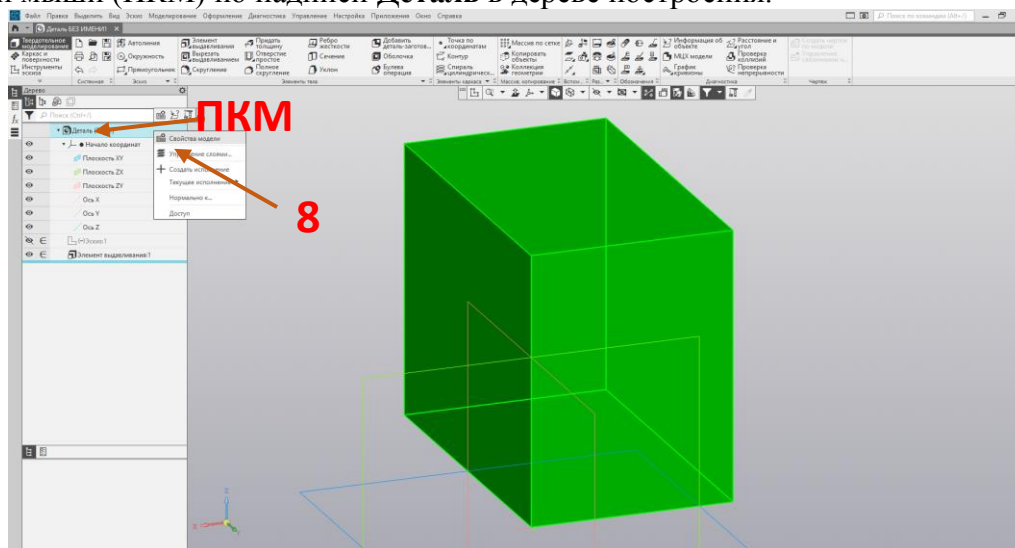
7. Закончить редактирование эскиза (повторно нажать на кнопку «эскиз»).
8. На панели **Элементы тела** выбрать **Элемент выдавливания**.
9. В окне **Параметры** в поле **Расстояние** установить значение 40 мм (высота параллелепипеда) и нажать кнопку **Создать** (Зелёная галочка), затем закрыть инструмент (красный крестик или клавиша Esc).



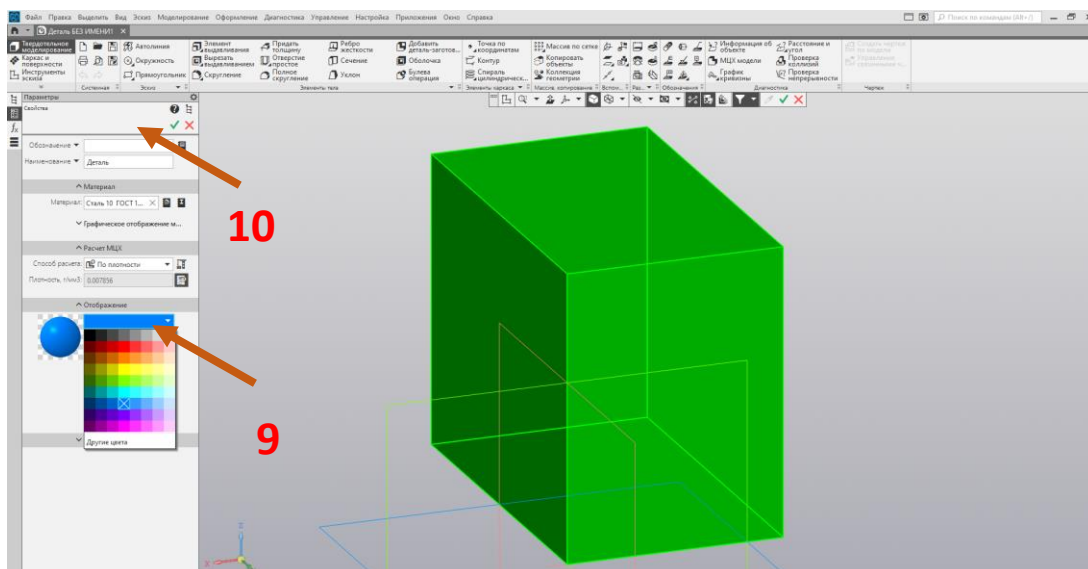
10. На экране программы должно появиться серое изображение параллелепипеда:



11. Чтобы изменить цвет параллелепипеда нужно вызвать свойства модели правой кнопкой мыши (ПКМ) по надписи **Деталь** в дереве построения.



В поле «отображение» выбрать интересующий цвет и нажать кнопку **Создать** (Зелёная галочка).



Трёхмерная модель параллелепипеда готова.

## Задача № 2 «Построение правильной пирамиды»

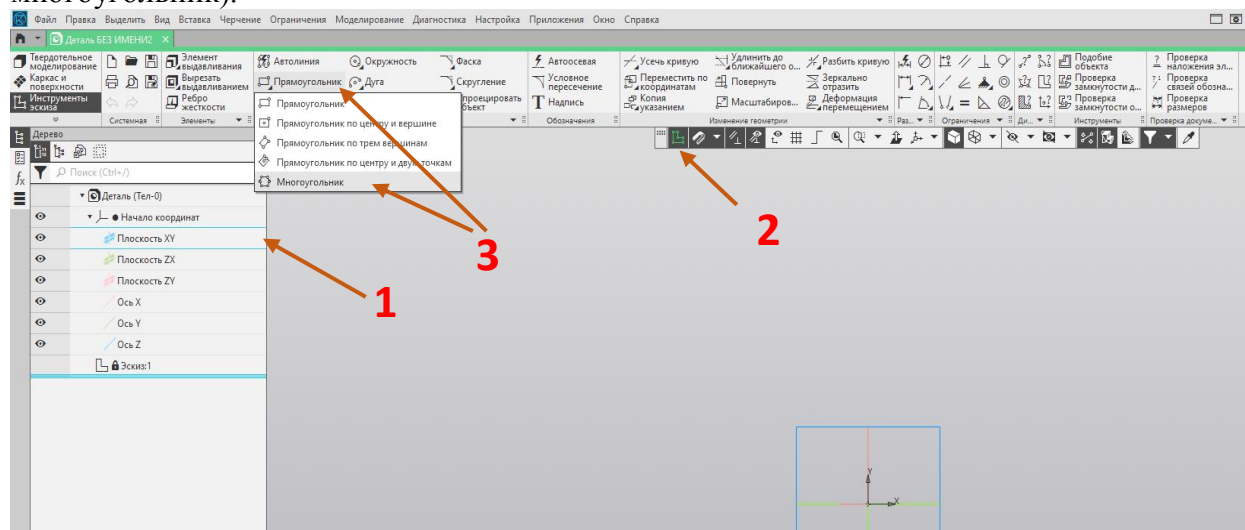
**Возраст:** 10-13 лет

**Условие:** построить трёхмерную модель правильной пирамиды в программе Компас 3D.

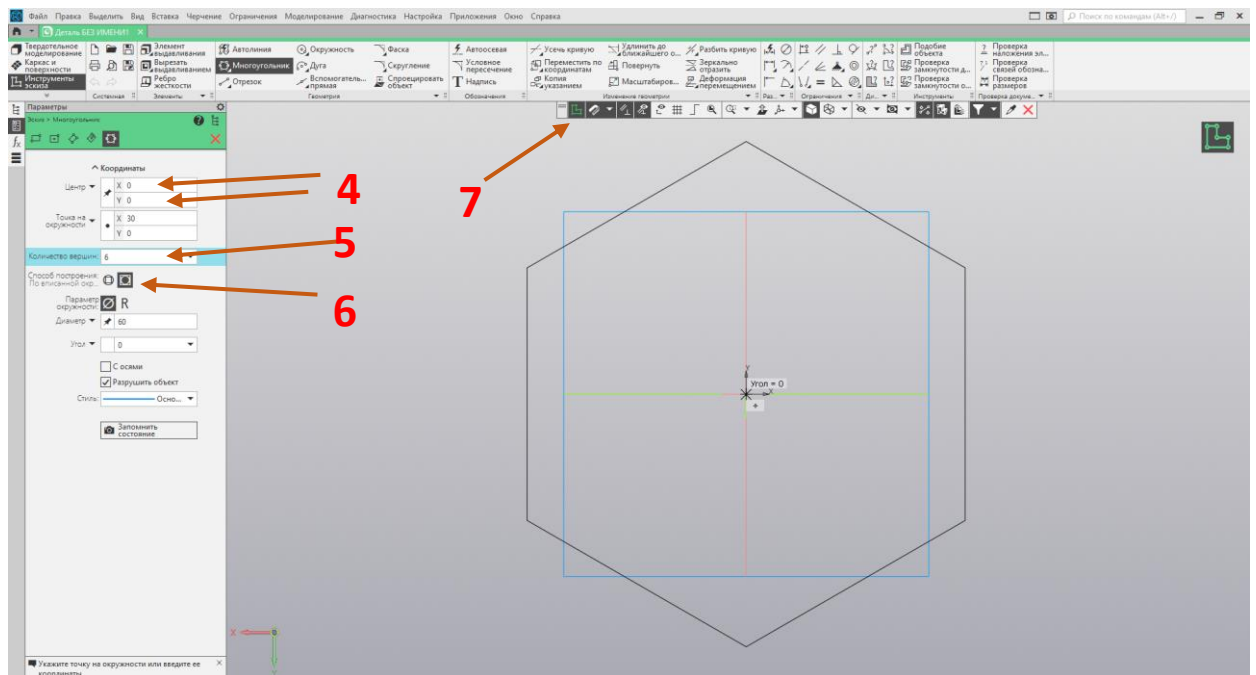
**Определения:** *Пирамида* называется правильной, если основанием её является правильный многоугольник, а вершина проецируется в центр основания.

**Порядок выполнения:**

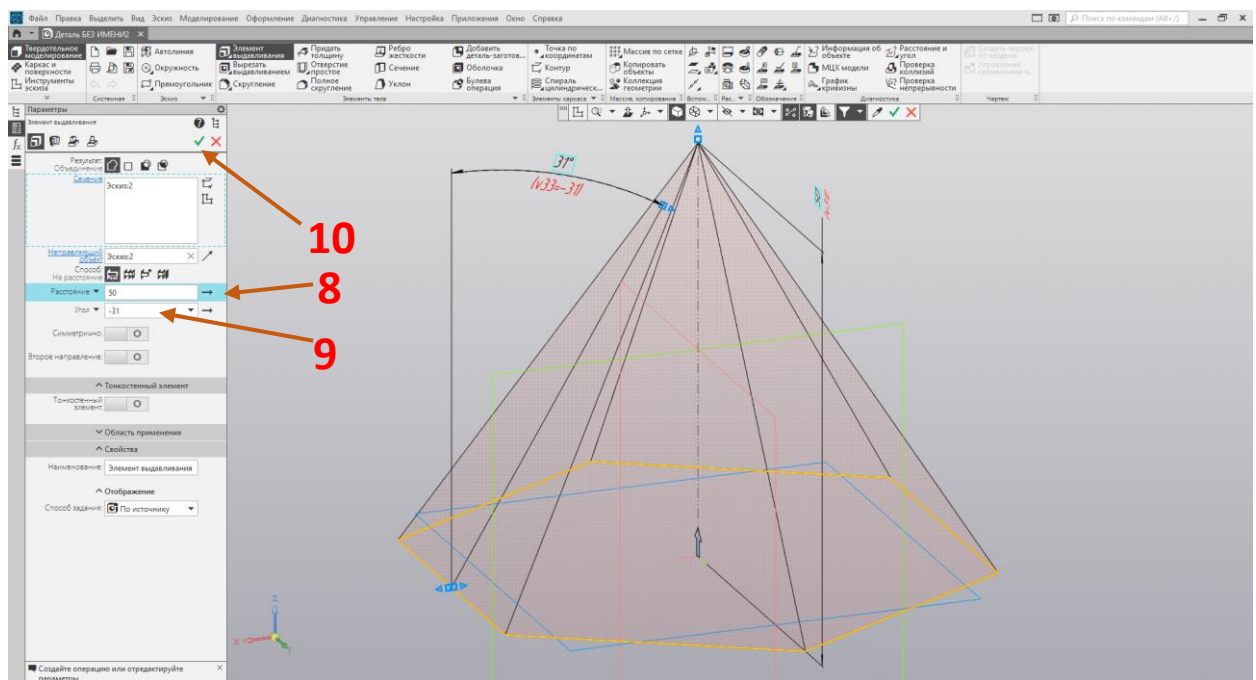
1. Запустить программу Компас 3D.
2. Выбрать создание детали (**Файл-Создать-Деталь**).
3. Выбрать в дереве модели плоскость XY.
4. Включить режим эскиз (кнопка панели управления).
5. На панели **Геометрия** активировать инструмент прямоугольник (в списке выбрать многоугольник).

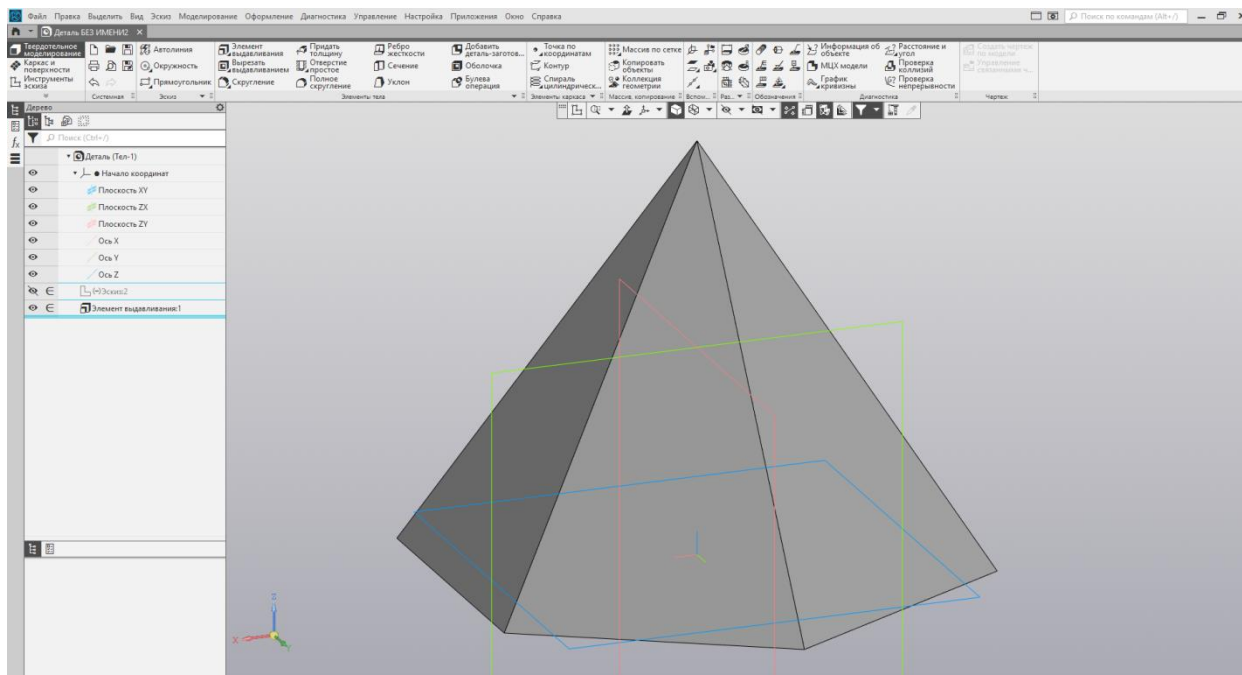


6. Ввести параметры: количество вершин 6; координаты центра - 0,0; способ построения - по вписанной окружности; диаметр окружности - 50 мм.



7. Закончить редактирование эскиза (повторно нажать на кнопку «эскиз»).
8. На панели **Элементы тела** выбрать **Элемент выдавливания**.
9. В окне **Параметры** установить значения: расстояние 50 мм (высота пирамиды); угол уклона -  $31^\circ$  (знак минус, не тире!) и нажать кнопку **Создать**.
10. На экране программы должно появиться изображение правильной пирамиды.





Трёхмерная модель правильной пирамиды готова.

### Задача № 3 «Построение цилиндра операцией выдавливания»

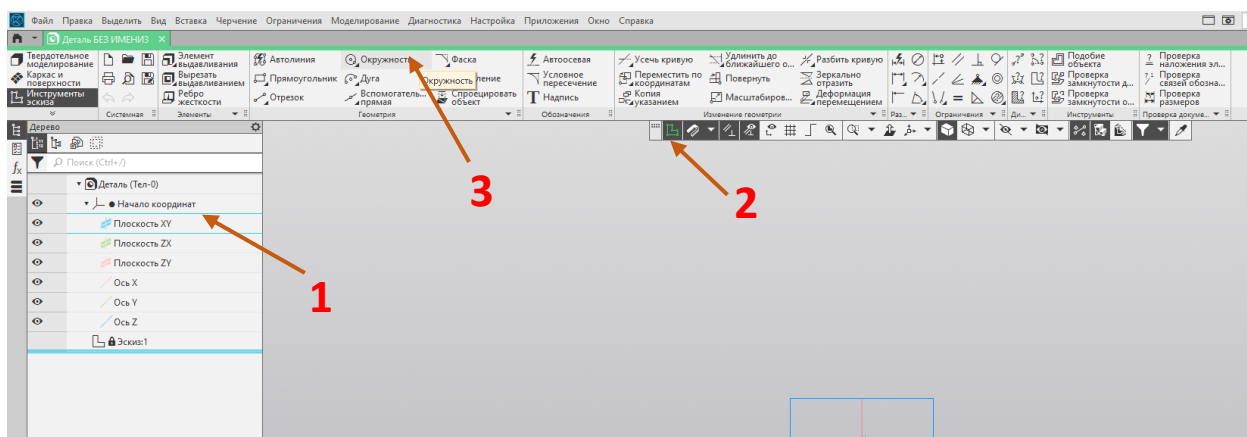
**Возраст:** 10-13 лет

**Условие:** построить трёхмерную модель цилиндра в программе Компас 3D.

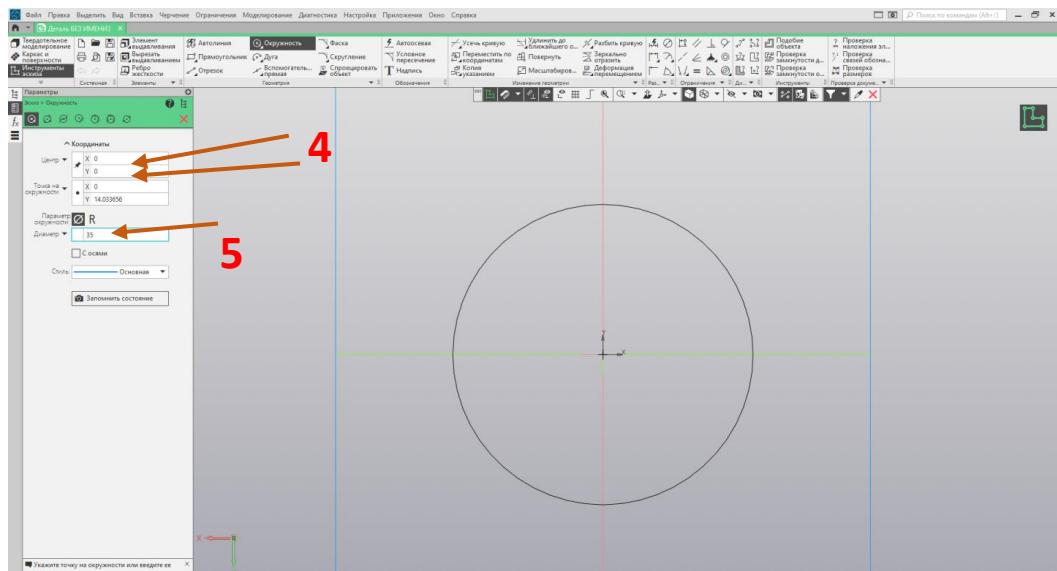
**Определения:** *Цилиндр* – геометрическое тело, ограниченное цилиндрической поверхностью и двумя параллельными плоскостями, пересекающими её.

**Порядок выполнения:**

1. Запустить программу Компас 3D.
2. Выбрать создание детали (**Файл-Создать-Деталь**).
3. Выбрать в дереве модели плоскость **XY**.
4. Включить режим эскиз (кнопка панели управления).
5. На панели **Геометрия** выбрать инструмент **Окружность**.

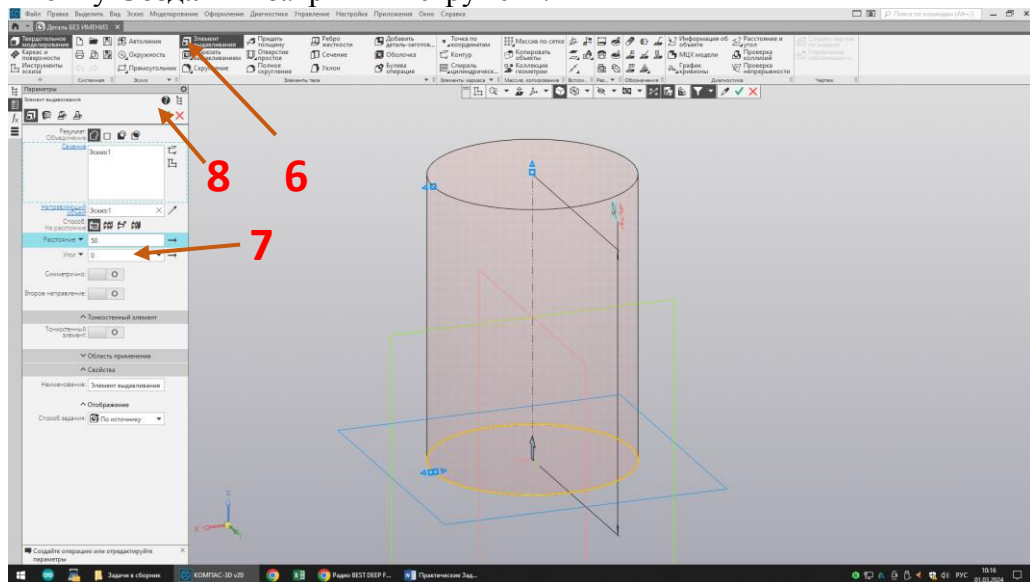


6. Ввести параметры: координаты центра - 0,0; диаметр окружности - 35 мм.

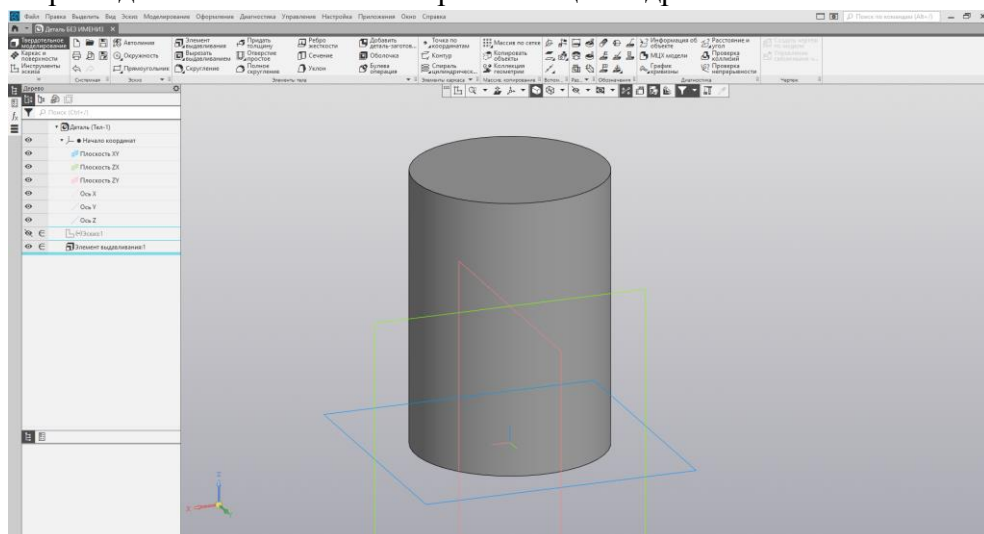


7. На панели **Элементы тела** выбрать **Элемент выдавливания**.

8. В окне **Параметры** установить значения: расстояние 50 мм (высота цилиндра) и нажать кнопку **Создать** и закрыть инструмент.



9. На экране должно появиться изображение цилиндра.



Трёхмерная модель цилиндра готова.



## Задача № 4 «Построение конуса операцией вращения»

**Возраст:** 10-13 лет

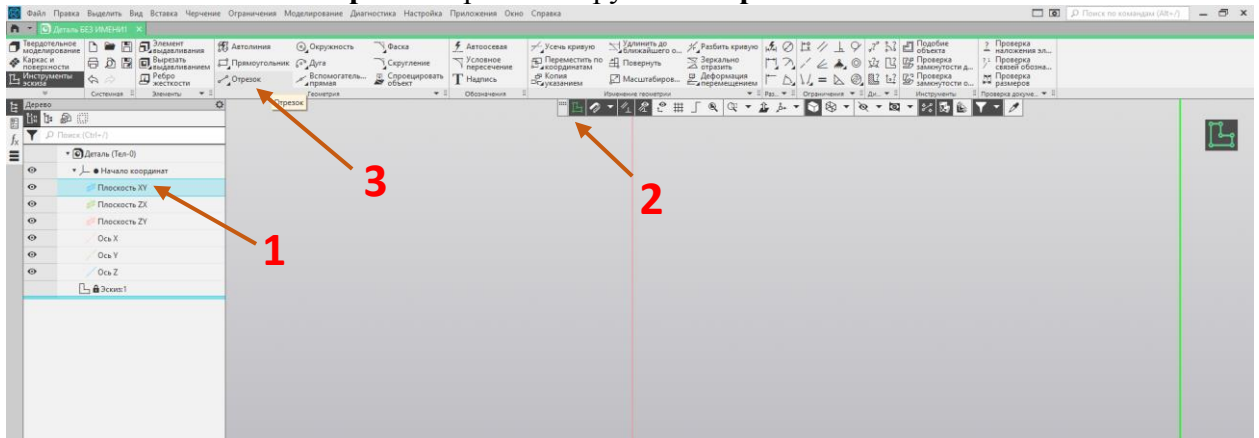
**Условие:** построить трехмерную модель конуса в программе Компас 3D.

**Определения:** *Конус* – это тело, полученное при вращении прямоугольного треугольника вокруг одного из его катетов.

*Операция вращения* – позволяет создавать детали методом вращения образующего контура вокруг осевой линии.

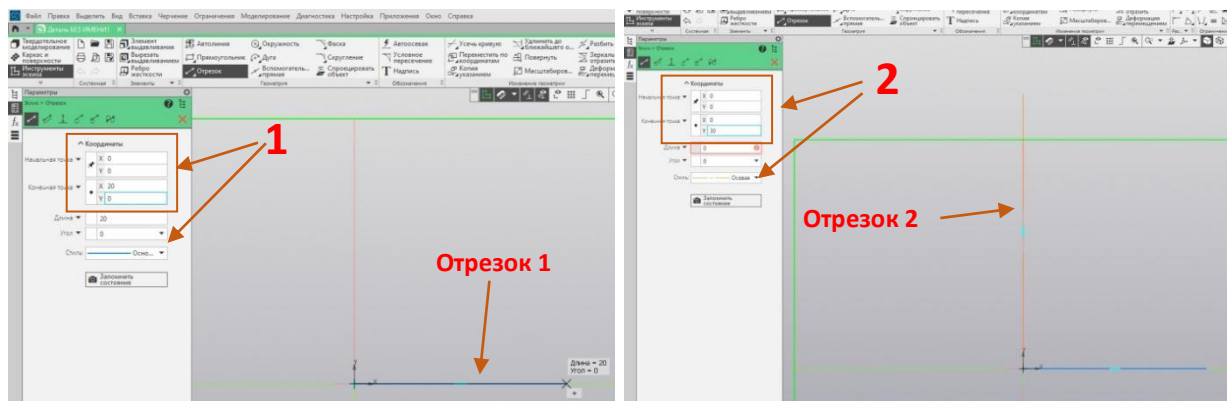
**Порядок выполнения:**

1. Запустить программу Компас 3D.
2. Выбрать создание детали (**Файл-Создать-Деталь**).
3. Выбрать в дереве модели плоскость **XU**.
4. Включить режим эскиз (кнопка панели управления).
5. На панели **Геометрия** выбрать инструмент **Отрезок**.

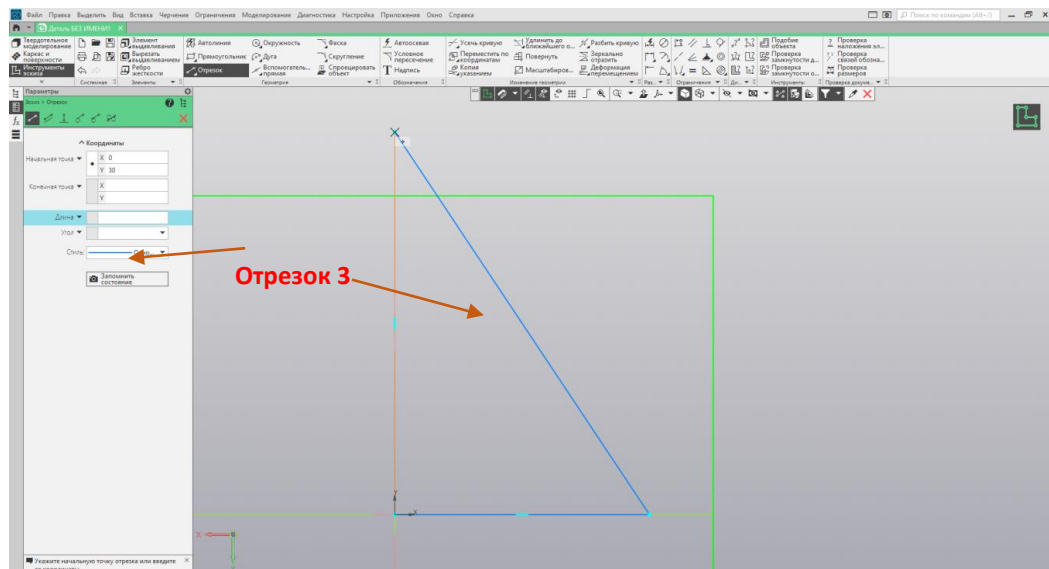


6. Ввести параметры 1 отрезка: координаты начала - 0,0; координаты конца - 20,0; стиль линии – **основная**.

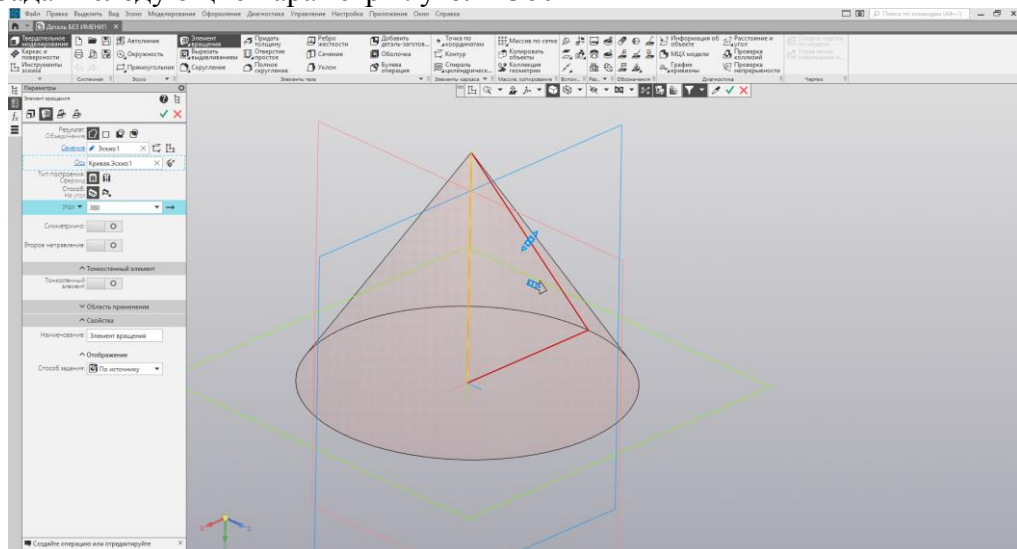
7. Ввести параметры 2 отрезка: координата начала – 0,0; координата конца – 0, 30; стиль линии – **осевая**.



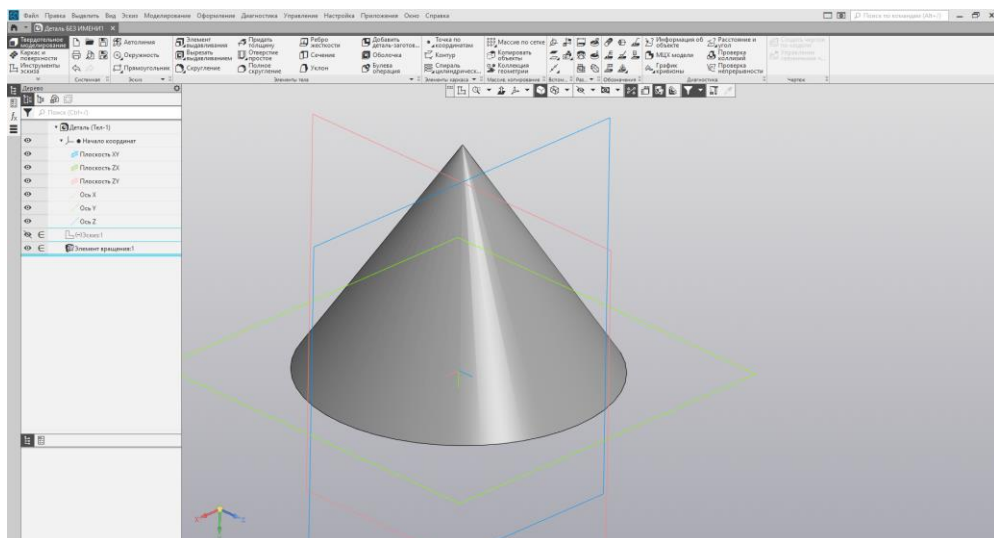
8. Соединить окончания отрезков 1 и 2 отрезком 3, стиль линии – **основная**.



9. Закончить редактирование эскиза (повторно нажать на кнопку «эскиз»).
10. На панели Элементы тела выбрать **Элемент вращения**.
11. Задать следующие параметры: угол -  $360^\circ$



12. Нажать кнопку **Создать**, на экране программы должно появиться изображение конуса.



Трёхмерная модель конуса построена.

## Задача № 5 «Построение тора операцияй вращения»

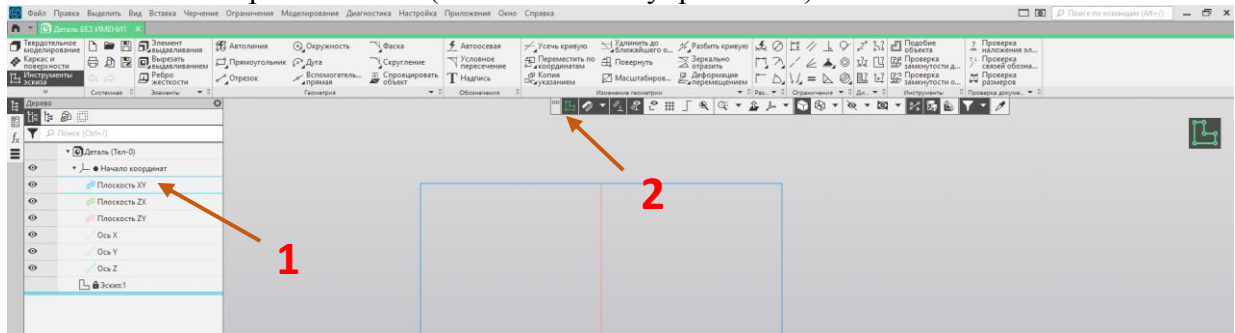
**Возраст:** 10-13 лет

**Условие:** построить трёхмерную модель тора в программе Компас 3D.

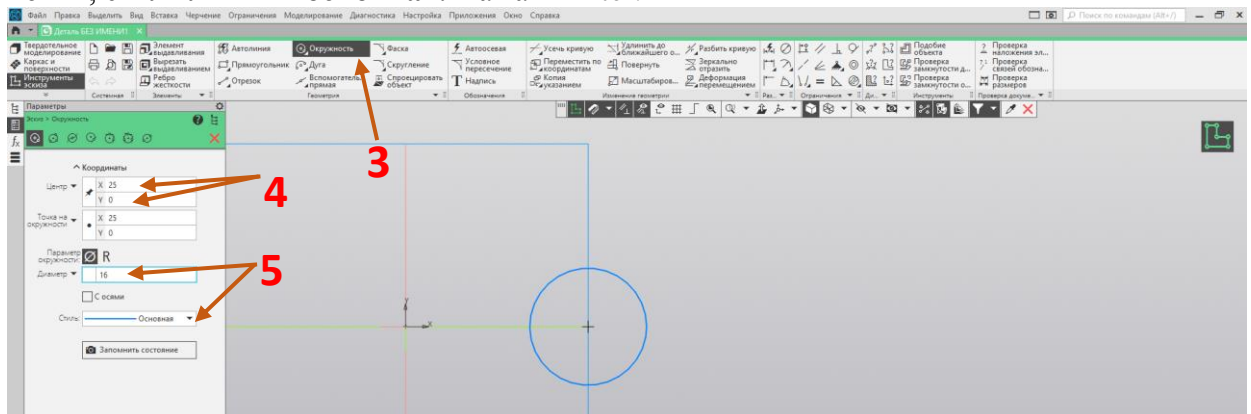
**Определения:** *Тор* – поверхность вращения, получаемая вращением образующей окружности вокруг оси, лежащей в плоскости этой окружности.

**Порядок выполнения:**

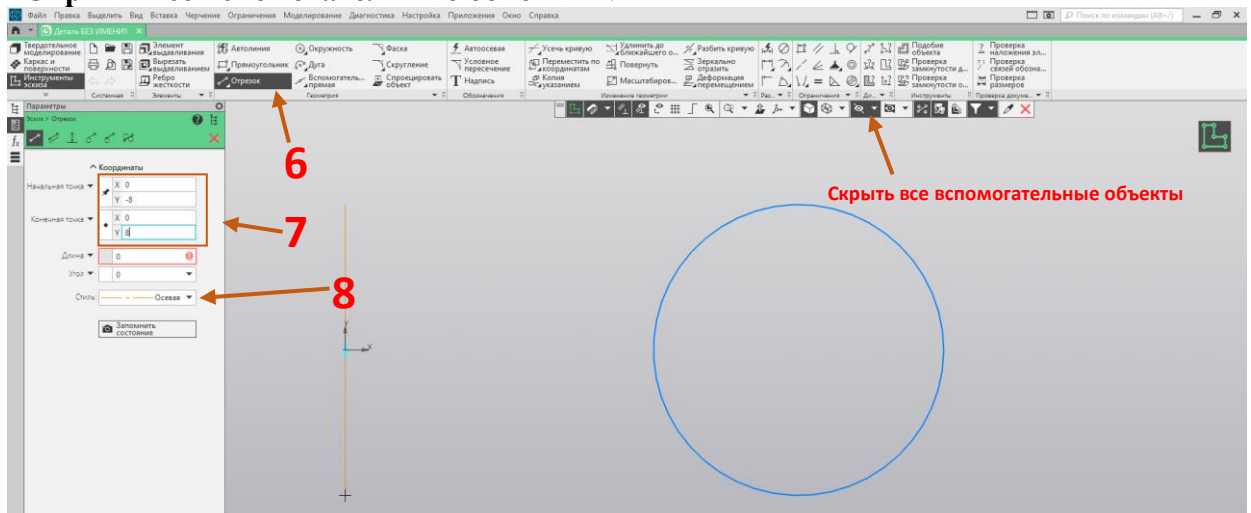
1. Запустить программу Компас 3D.
2. Выбрать создание детали (**Файл-Создать-Деталь**).
3. Выбрать в дереве модели плоскость **XU**.
4. Включить режим эскиз (кнопка панели управления).



5. На панели **Геометрия** выбрать инструмент **Окружность**.
6. Ввести параметры окружности: координаты центра – 25, 0; диаметр окружности – 16 мм, стиль линии – **основная**. Нажать Enter.

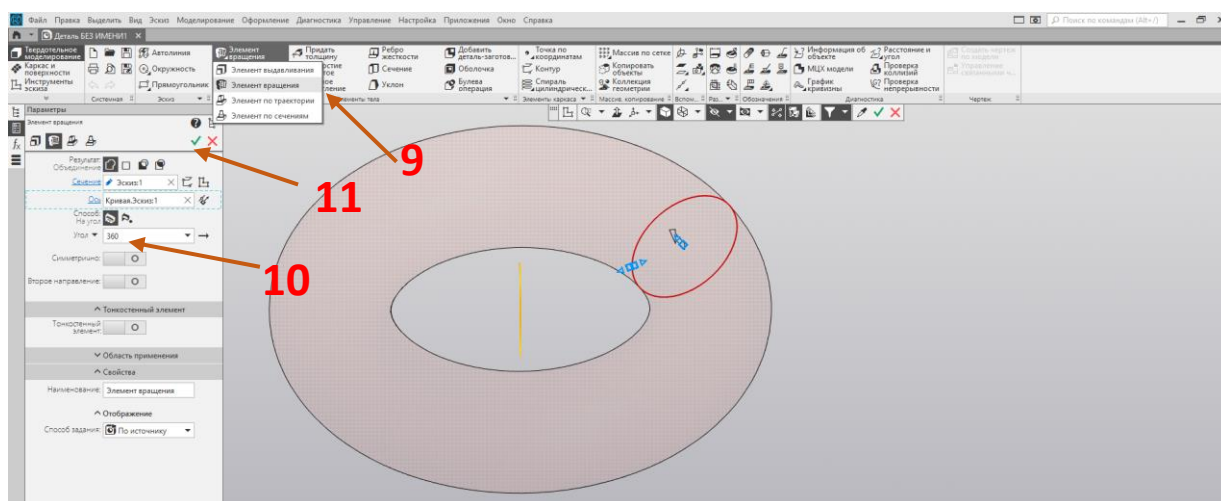


7. Начертить отрезок с параметрами: координаты начала – 0,-8; координаты конца – 0, 8; стиль линии – **осевая**. Чтобы отрезок было видно нажмите на пиктограмму «Скрыть все вспомогательные объекты».

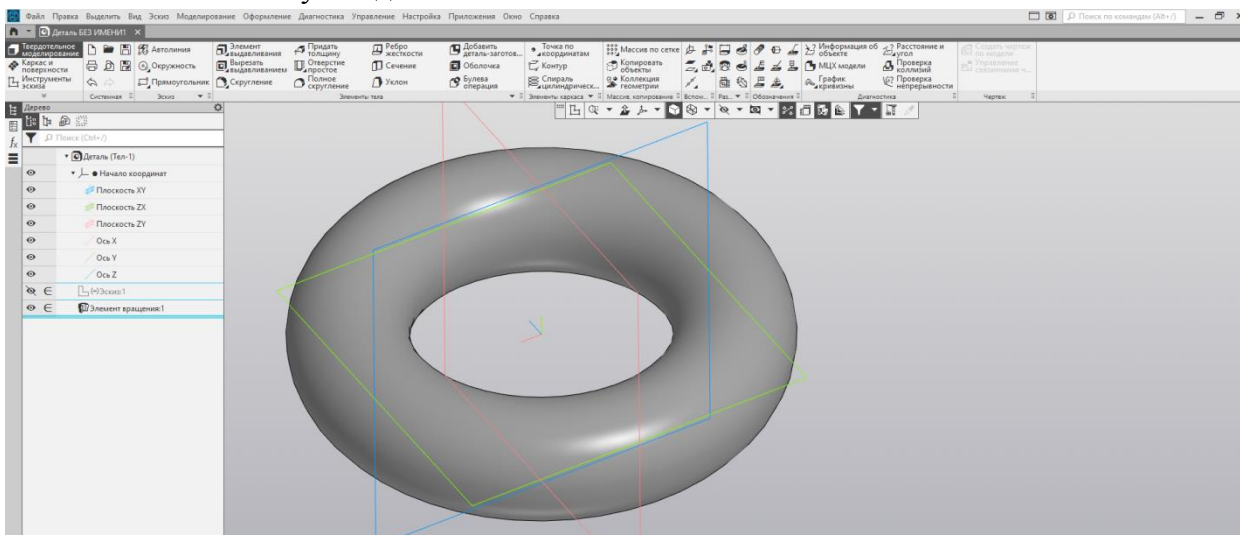




8. Закончить редактирование эскиза (повторно нажать на кнопку «эскиз»).
9. На панели Элементы тела выбрать **Элемент вращения**.
10. Задать следующие параметры: угол -  $360^\circ$  и нажать кнопку **Создать**.
11. На экране программы должно появиться изображение тора.



12. Нажать кнопку **Создать**.



Трёхмерная модель тора построена.

## Задача № 6 «Трёхмерное моделирование сложных тел. Построение составной пирамиды»

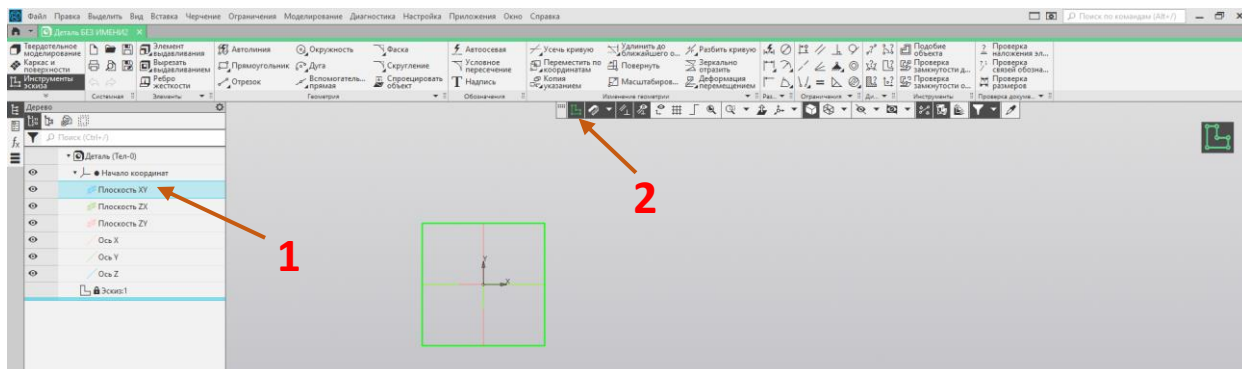
**Возраст:** 10-13 лет

**Условие:** построить трёхмерную модель составной пирамиды в программе Компас 3D.

**Определения:** *Составная пирамида* – геометрическое тело, представляющее собой составленные по вертикали цилиндры, причем ось вращения всех цилиндров лежит на единой прямой, а диаметр цилиндров уменьшается с высотой.

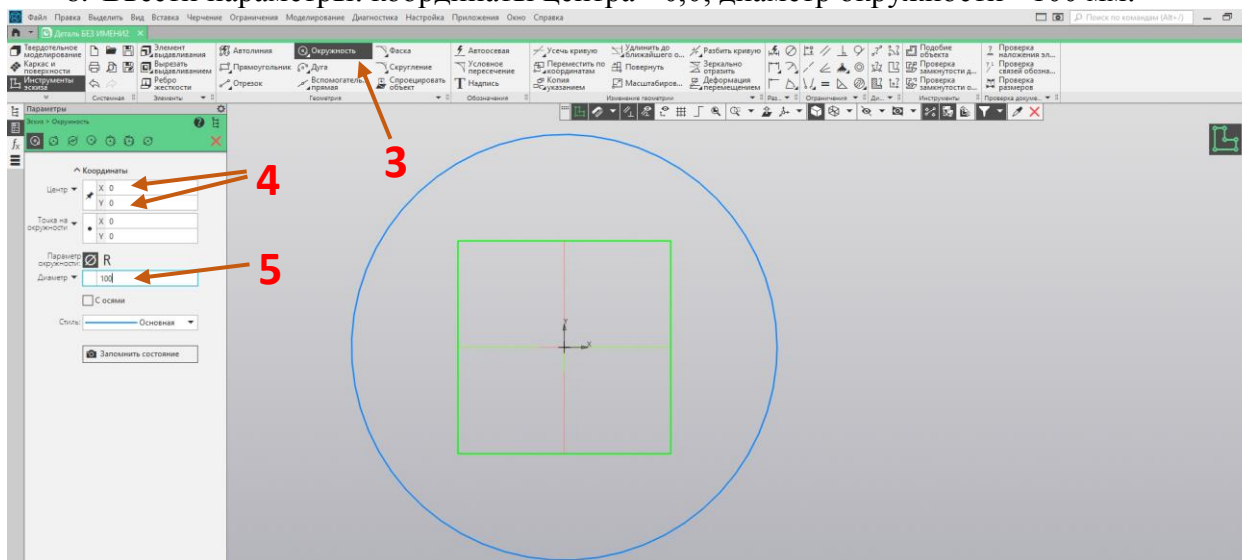
**Порядок выполнения:**

1. Запустить программу Компас 3D.
2. Выбрать создание детали (**Файл-Создать-Деталь**).
3. Выбрать в дереве модели плоскость **XY**.
4. Включить режим эскиз (кнопка панели управления).



5. На панели **Геометрия** выбрать инструмент **Окружность**.

6. Ввести параметры: координаты центра - 0,0; диаметр окружности - 100 мм.

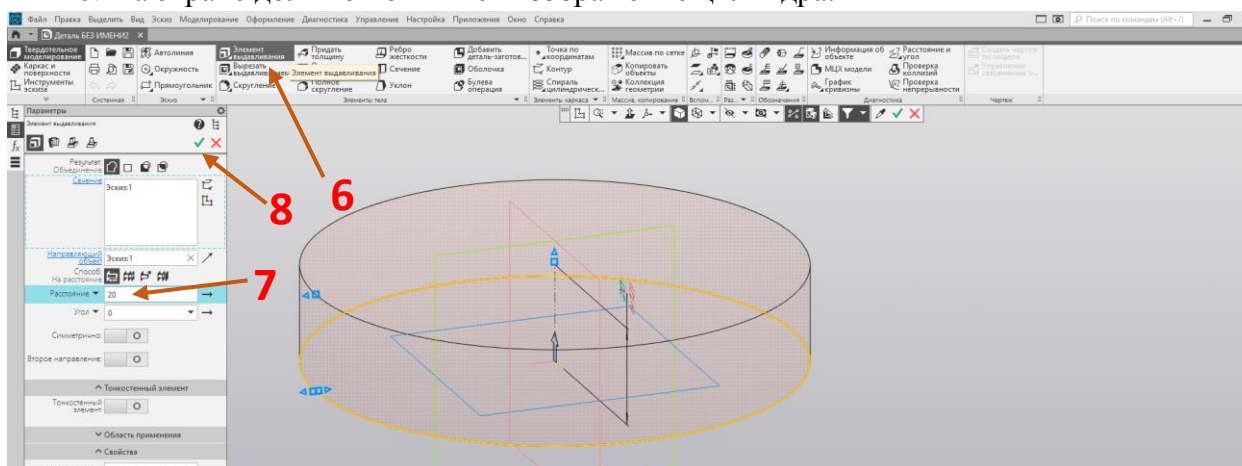


7. Закончить редактирование эскиза (повторно нажать на кнопку «эскиз»).

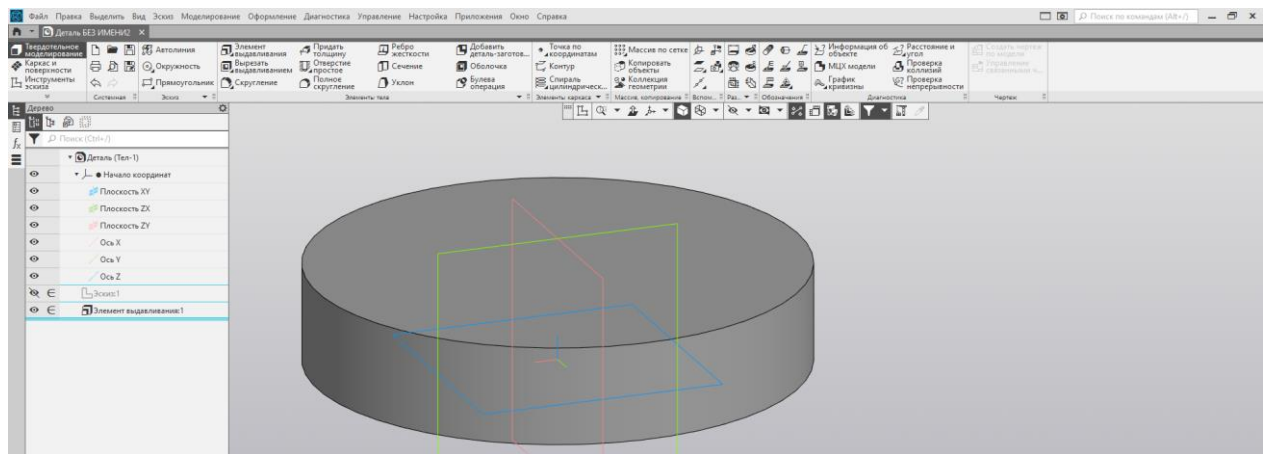
8. На панели **Элементы тела** выбрать **Элемент выдавливания**.

9. В окне **Параметры** на вкладке **Элемент выдавливания** установить параметры: расстояние 20 мм (высота цилиндра) и нажать кнопку **Создать**.

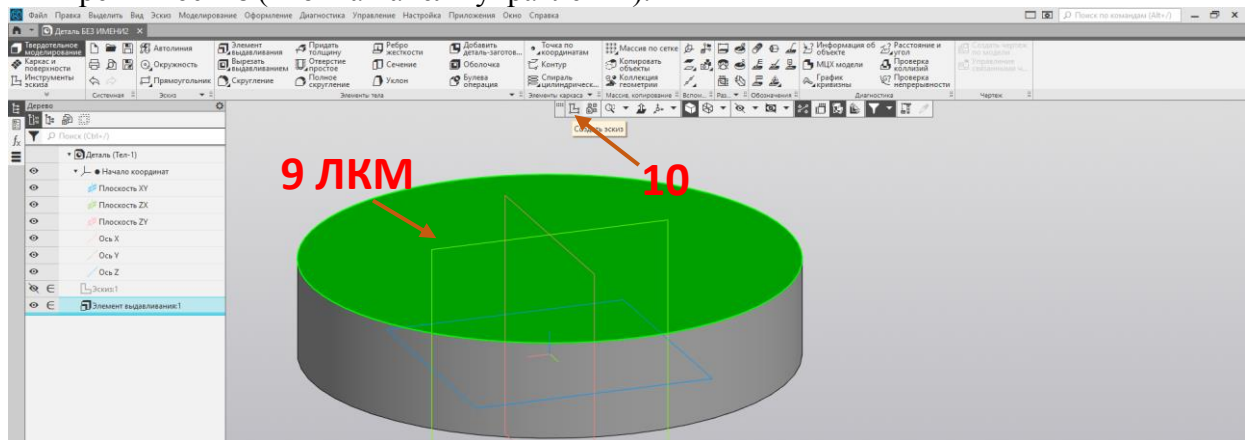
10. На экране должно появиться изображение цилиндра.



11. Нажать Enter и закрыть инструмент

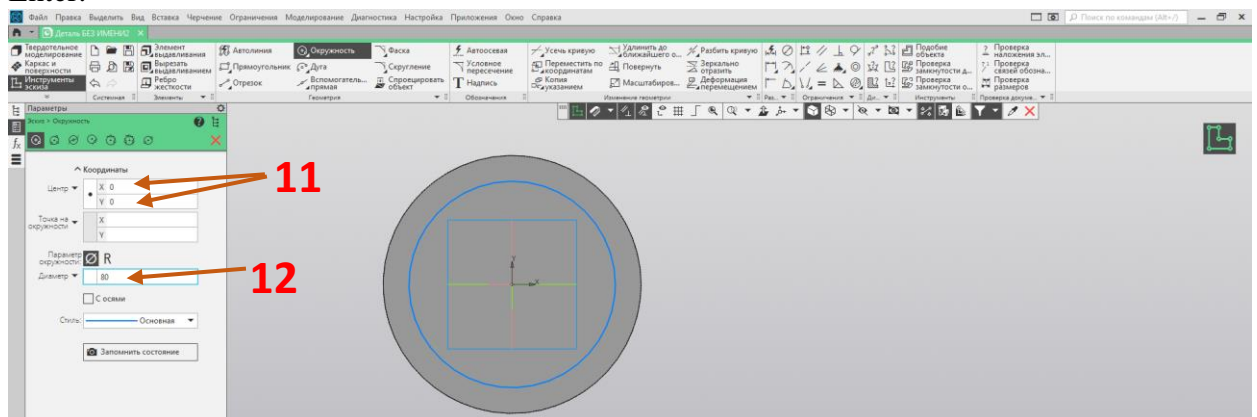


12. Выбрать верхнюю грань цилиндра: щелкнуть левой кнопкой мыши (ЛКМ) и включить режим эскиз (кнопка панели управления).



13. На панели Геометрия выбрать инструмент Окружность.

14. Ввести параметры: координаты центра - 0,0; диаметр окружности -80 мм. Нажать Enter.



15. На панели Элементы тела выбрать Элемент выдавливания.

16. В окне Параметры установить значение расстояния 20 мм и нажать кнопку Создать.

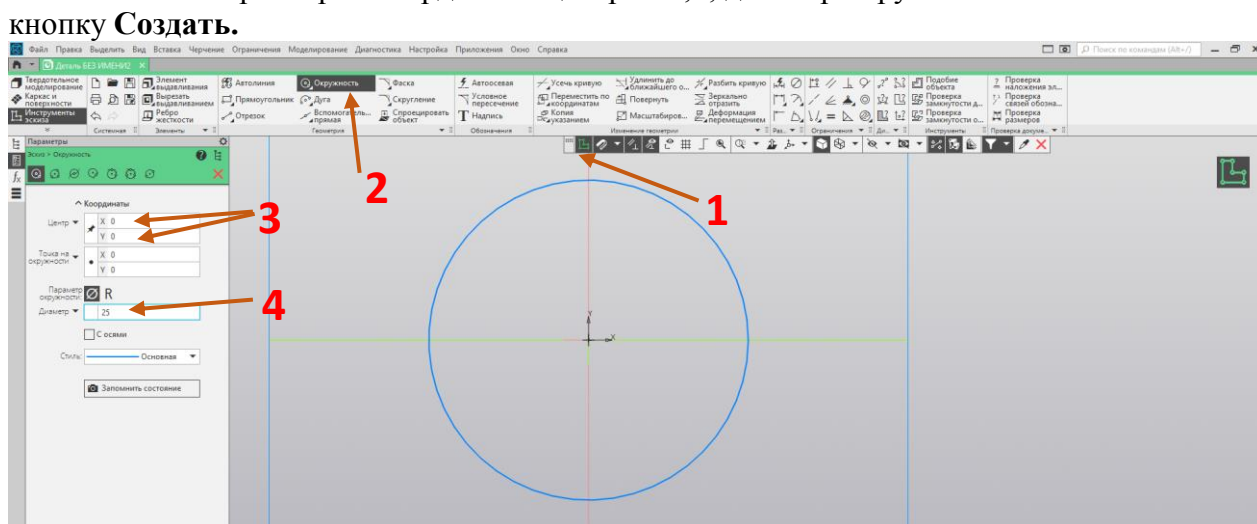




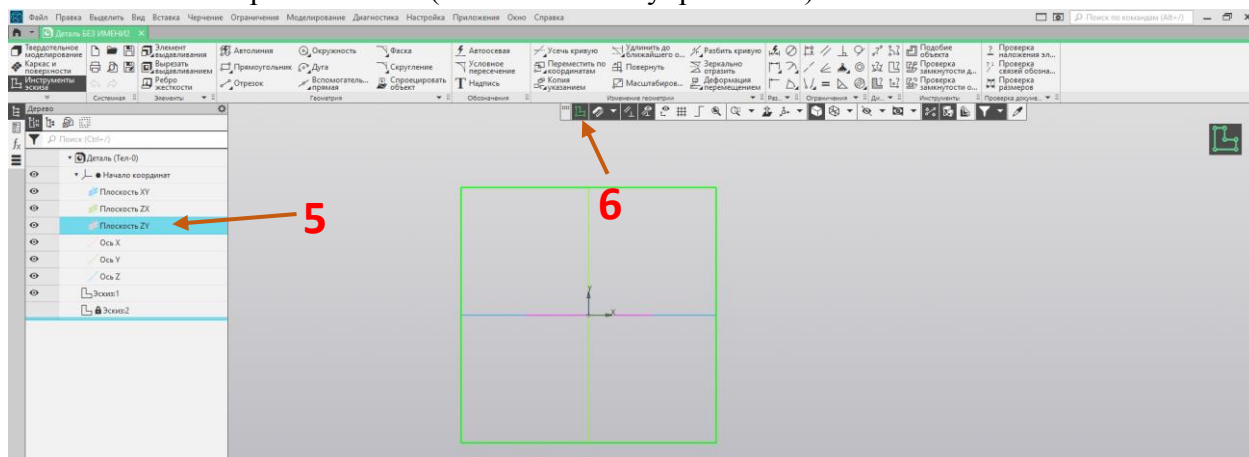
**Определения:** *Кинематический способ задания поверхностей* – основан на непрерывном перемещении образующего контура (сечения) в пространстве по определенному закону (траектории).

**Порядок выполнения:**

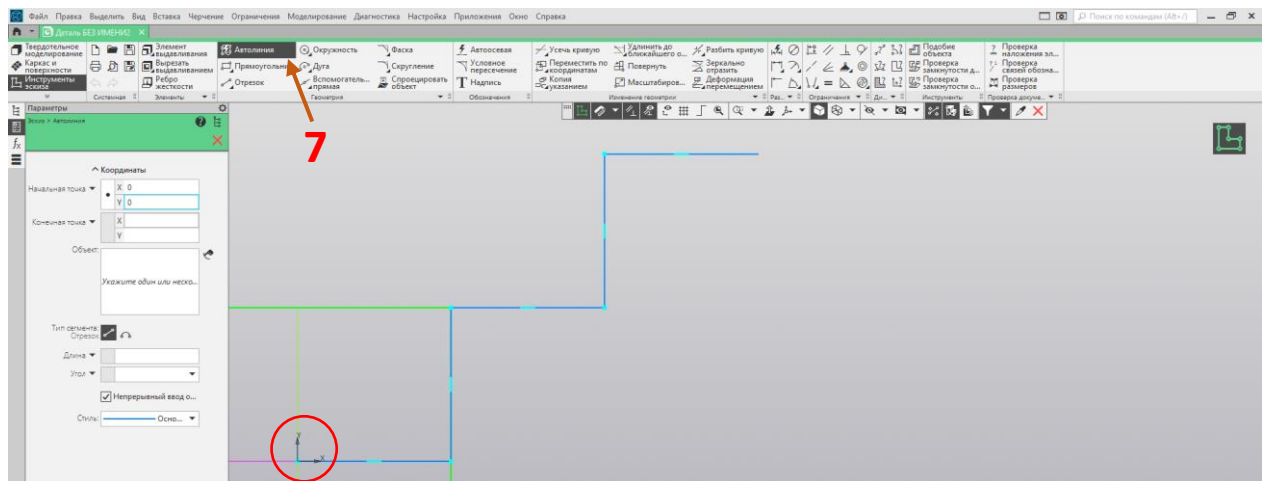
1. Запустить программу Компас 3D.
2. Выбрать создание детали (**Файл-Создать-Деталь**).
3. Выбрать в дереве модели плоскость **ZX**.
4. Включить режим эскиз (кнопка панели управления).
5. На панели **Геометрия** выбрать инструмент **Окружность**.
6. Ввести параметры: координаты центра - 0,0; диаметр окружности - 25 мм. Нажать кнопку **Создать**.



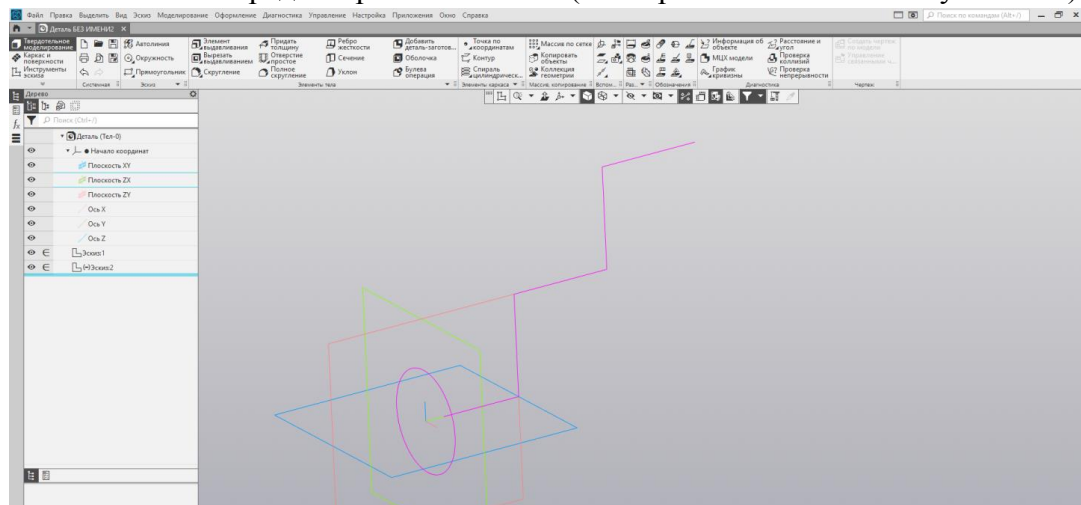
7. Закончить редактирование эскиза (повторно нажать на кнопку «эскиз»).
8. Выбрать в дереве модели плоскость **ZY**.
9. Включить режим эскиз (кнопка панели управления).



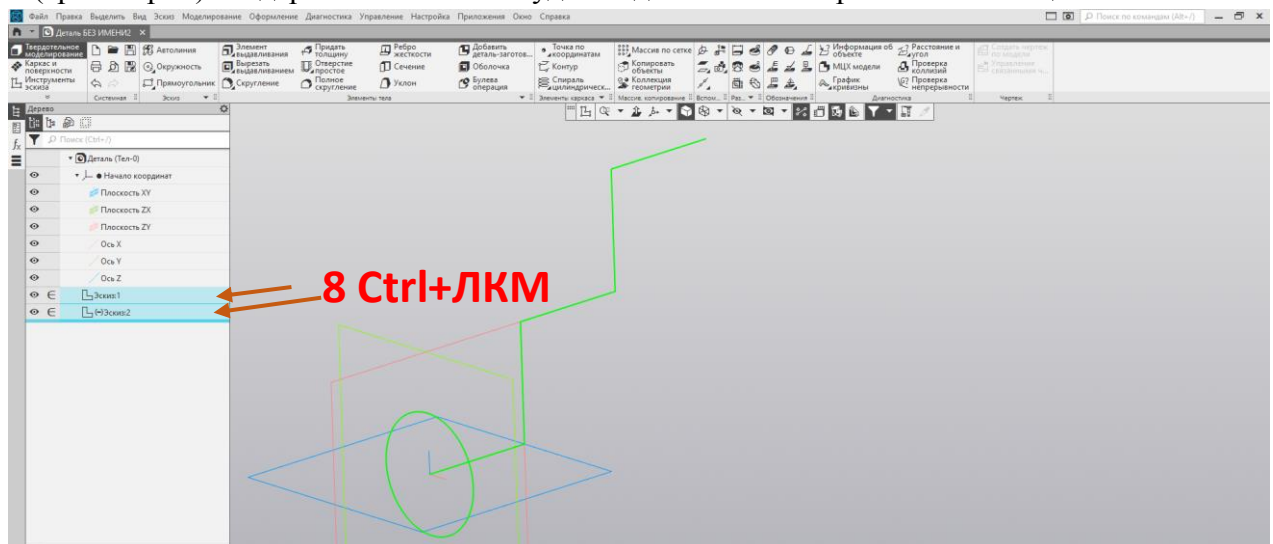
10. На панели **Геометрия** выбрать инструмент **Автолиния**.
11. Построить последовательно 5 отрезков длинами 50 мм в виде ступенок как показано на картинке ниже. Начать построение отрезков нужно с начальной точки 0,0 локальной системы координат (отмечена окружностью). Нажать кнопку **Создать**.



12. Закончить редактирование эскиза (повторно нажать на кнопку «эскиз»).

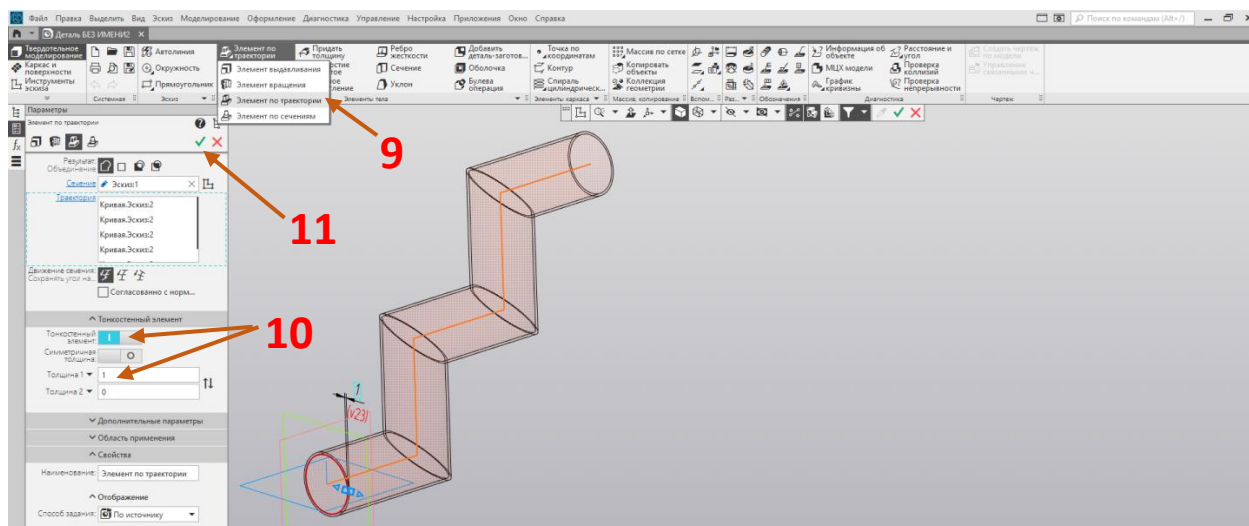


13. Выделить с зажатой клавишей **Ctrl** в дереве модели эскиз 1 (окружность) и эскиз 2 (траектория). Содержимое эскизов будет подсвечиваться ярким зеленым цветом.



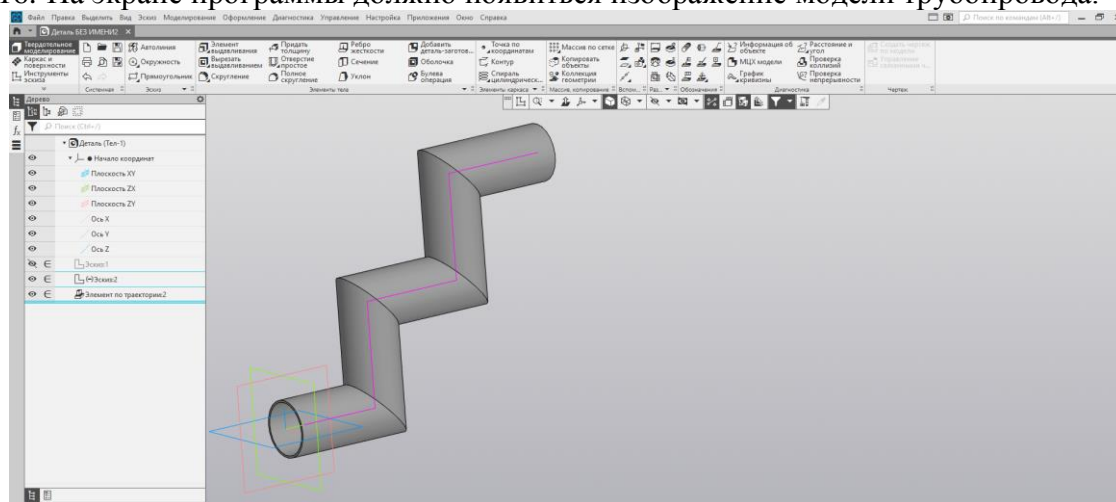
14. На панели **Элементы тела** выбрать **Элемент по траектории**.

15. Инструмент **Элемент по траектории** автоматически определит Эскиз 1 как сечение (рабочий контур), а Эскиз 2 как траекторию:



В окне параметров включить переключатель **Тонкостенный элемент**, **Толщина 1** -1 мм и нажать кнопку **Создать**.

16. На экране программы должно появиться изображение модели трубопровода.



Трёхмерная модель трубопровода построена.

## Задача № 8: «Трёхмерное моделирование тела с применением кинематической операции «Элемент по сечениям»»

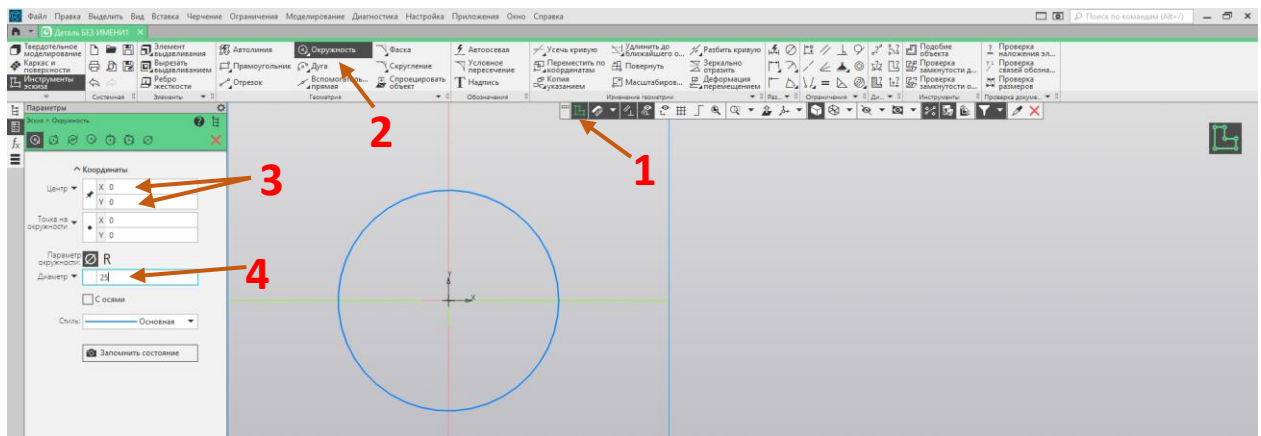
**Возраст:** 10-13 лет

**Условие:** построить трёхмерную модель вазы в программе Компас 3D.

**Определения:** **Сечения** – изображение фигуры, получающейся при мысленном рассечении предмета одной или несколькими плоскостями. На сечении показывается только то, что получается непосредственно в секущей плоскости.

**Порядок выполнения:**

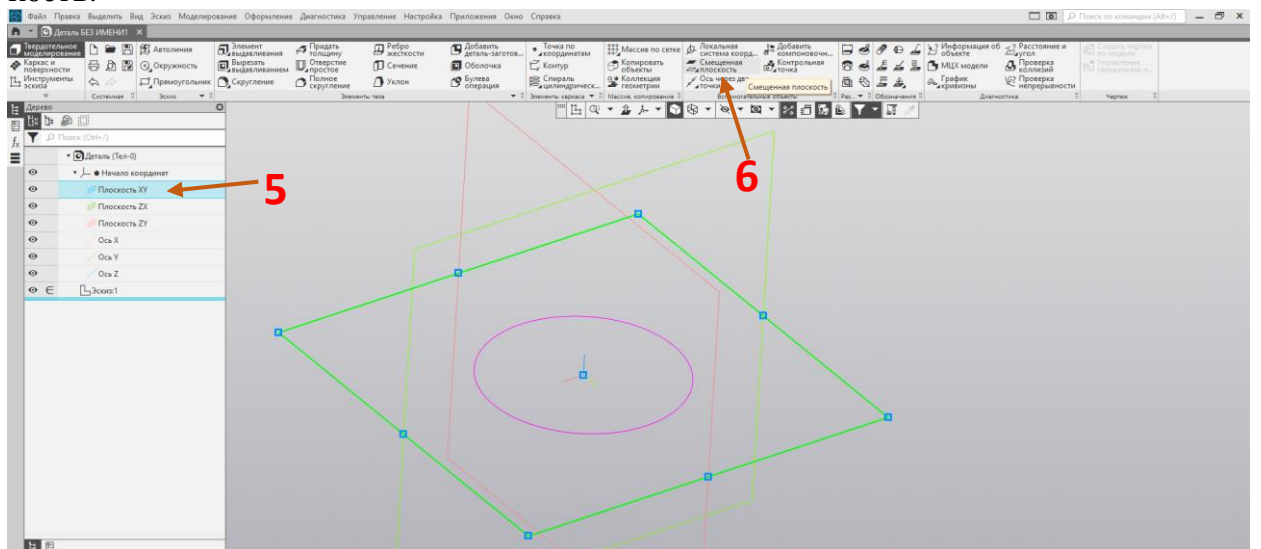
1. Запустить программу Компас 3D.
2. Выбрать создание детали (**Файл-Создать-Деталь**).
3. Выбрать в дереве модели плоскость **XY**.
4. Включить режим эскиз (кнопка панели управления).
5. На панели **Геометрия** выбрать инструмент **Окружность**.
6. Ввести параметры: координаты центра - 0,0; диаметр окружности - 25 мм. Нажать кнопку **Создать**.



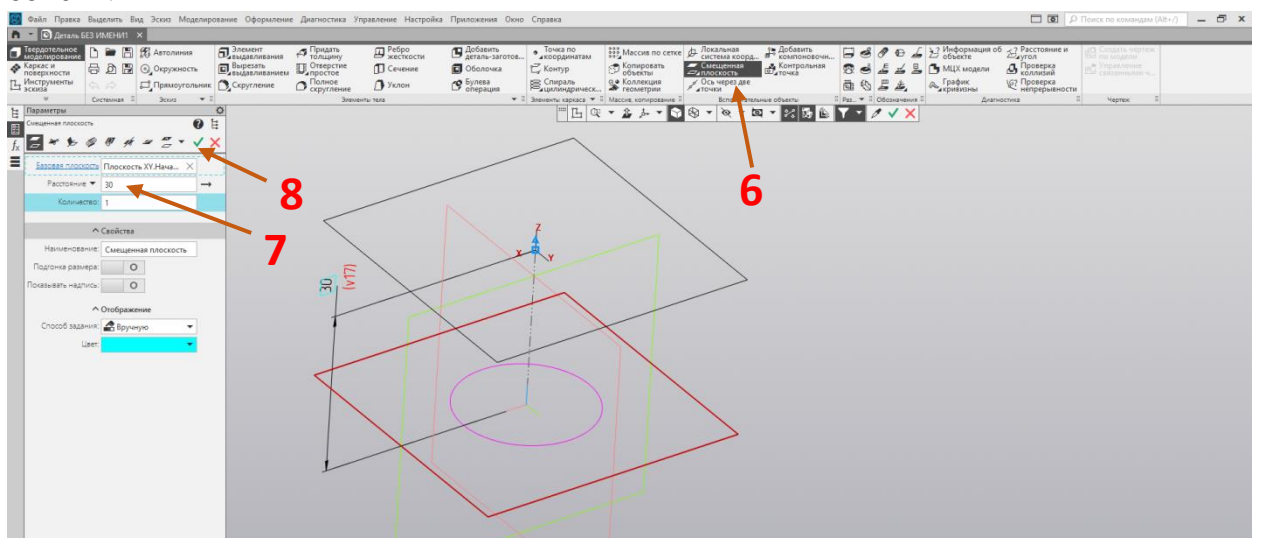
7. Закончить редактирование эскиза (повторно нажать на кнопку «эскиз»).

8. Выбрать в дереве модели плоскость XY.

9. Выбрать во вкладке **Вспомогательная геометрия** инструмент **Смещенная плоскость**.

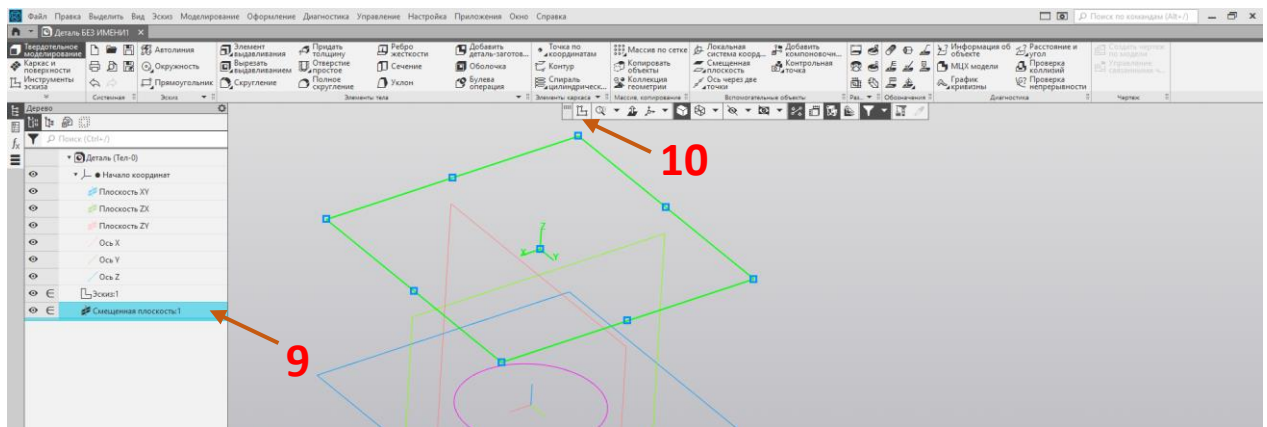


10. Установить следующие параметры: расстояние – 30 мм. Нажать кнопку **Создать объект**.



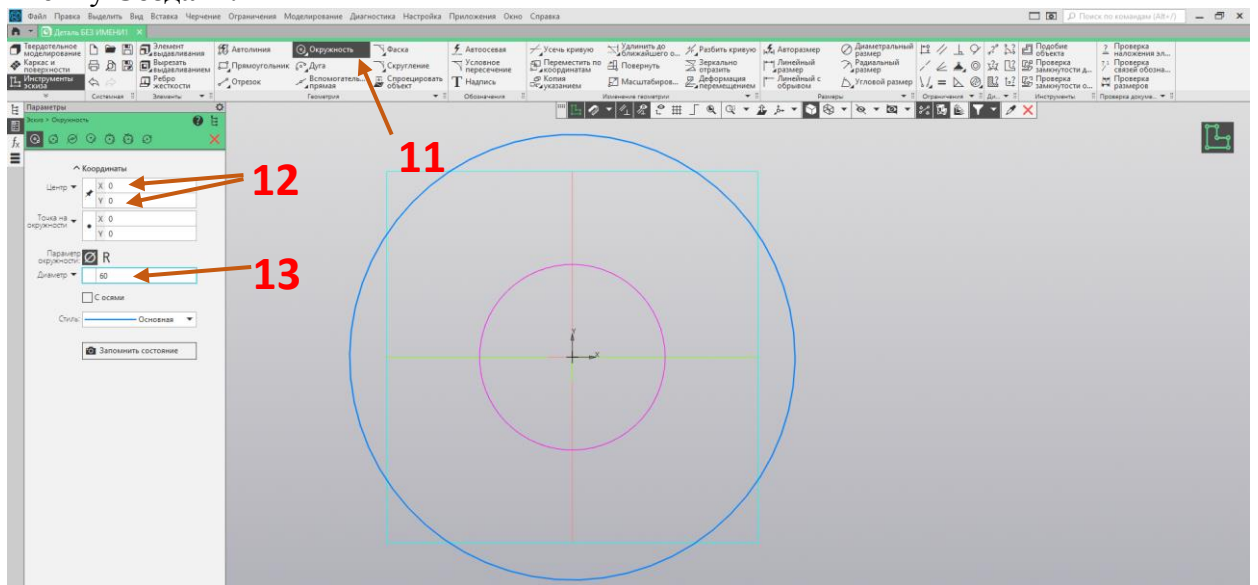
11. Выбрать в дереве модели Смещенную плоскость 1 и включить режим эскиз.





12. На панели **Геометрия** выбрать инструмент **Окружность**.

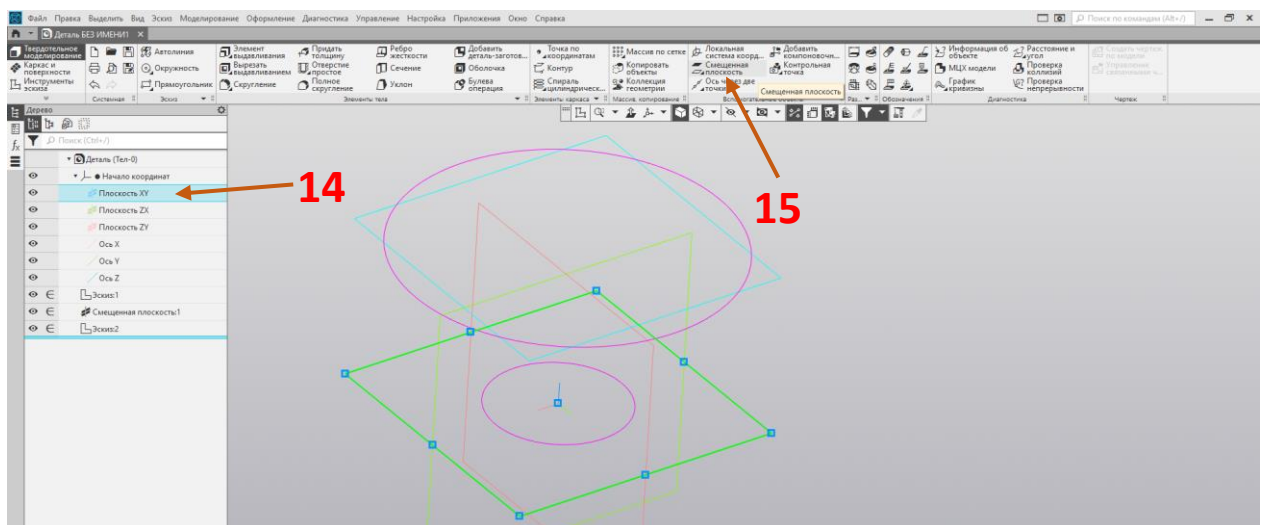
13. Ввести параметры: координаты центра - 0,0; диаметр окружности - 60 мм. Нажать кнопку **Создать**.

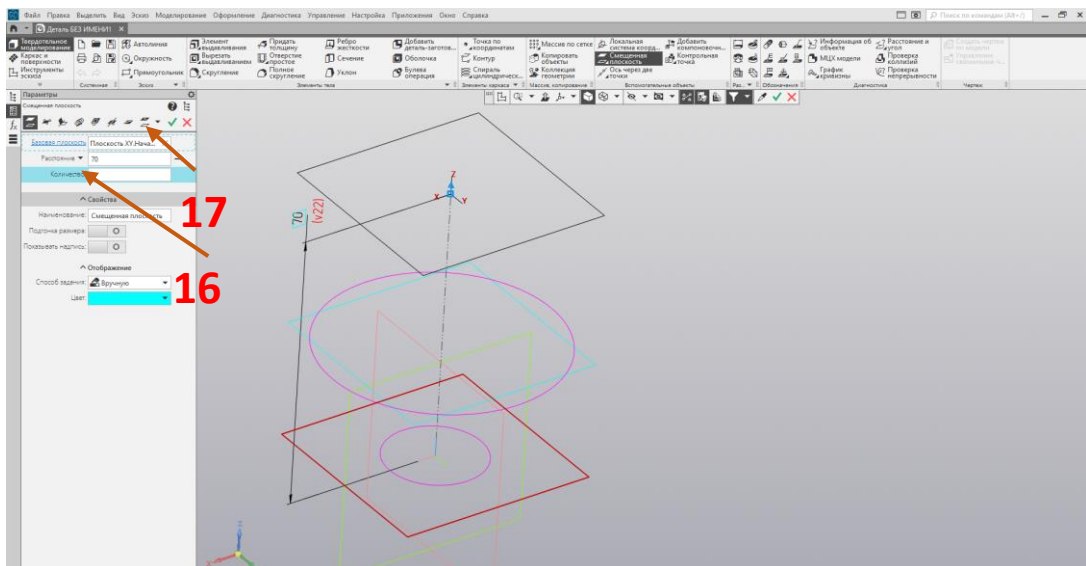


14. Закончить редактирование эскиза (повторно нажать на кнопку «эскиз»).

15. Выбрать в дереве модели плоскость **XU**.

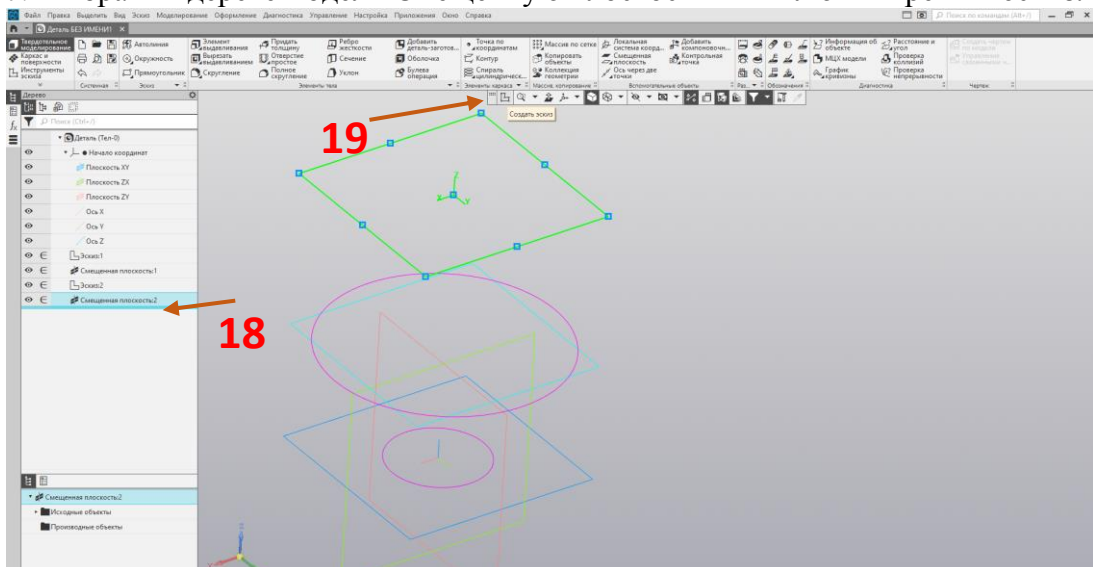
16. Выбрать во вкладке **Вспомогательная геометрия** инструмент **Смещенная плоскость**. Установить следующие параметры: расстояние – 70 мм.





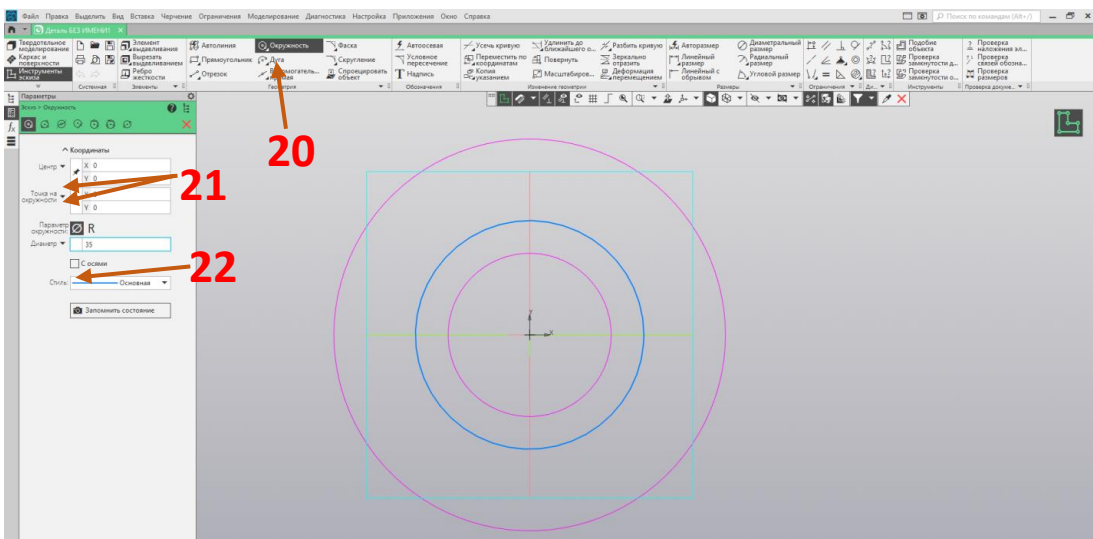
Нажать кнопку **Создать объект**.

17. Выбрать в дереве модели **Смещенную плоскость 2** и включить режим эскиза.

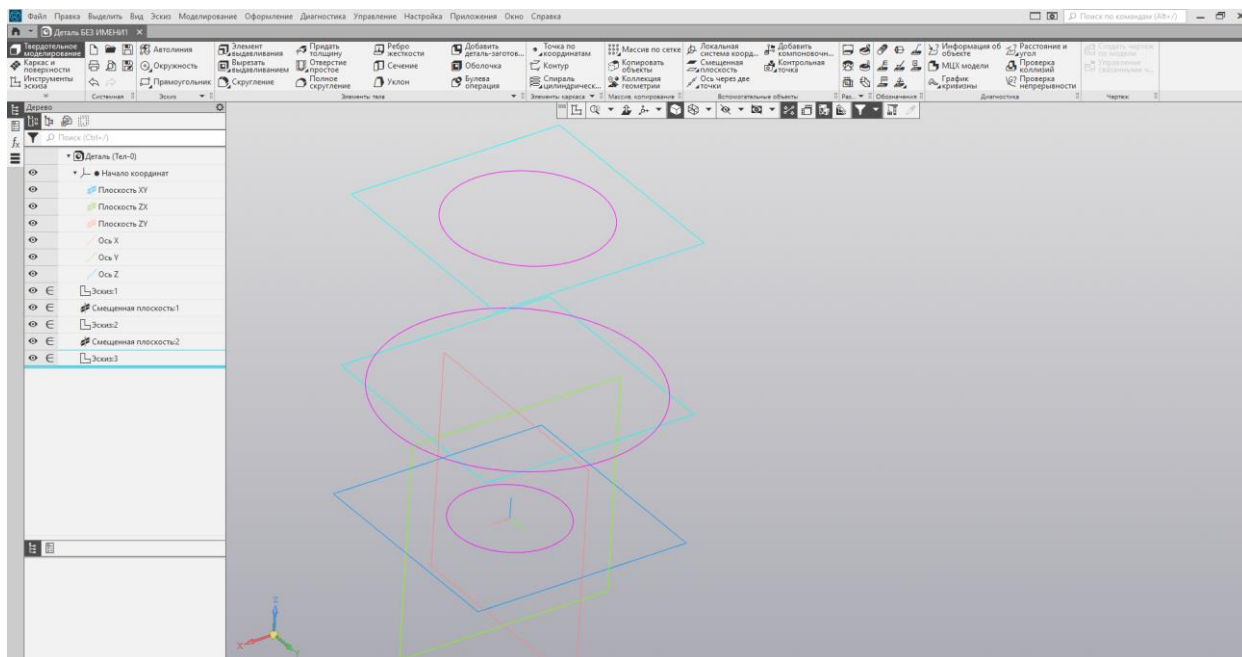


18. На панели **Геометрия** выбрать инструмент **Окружность**.

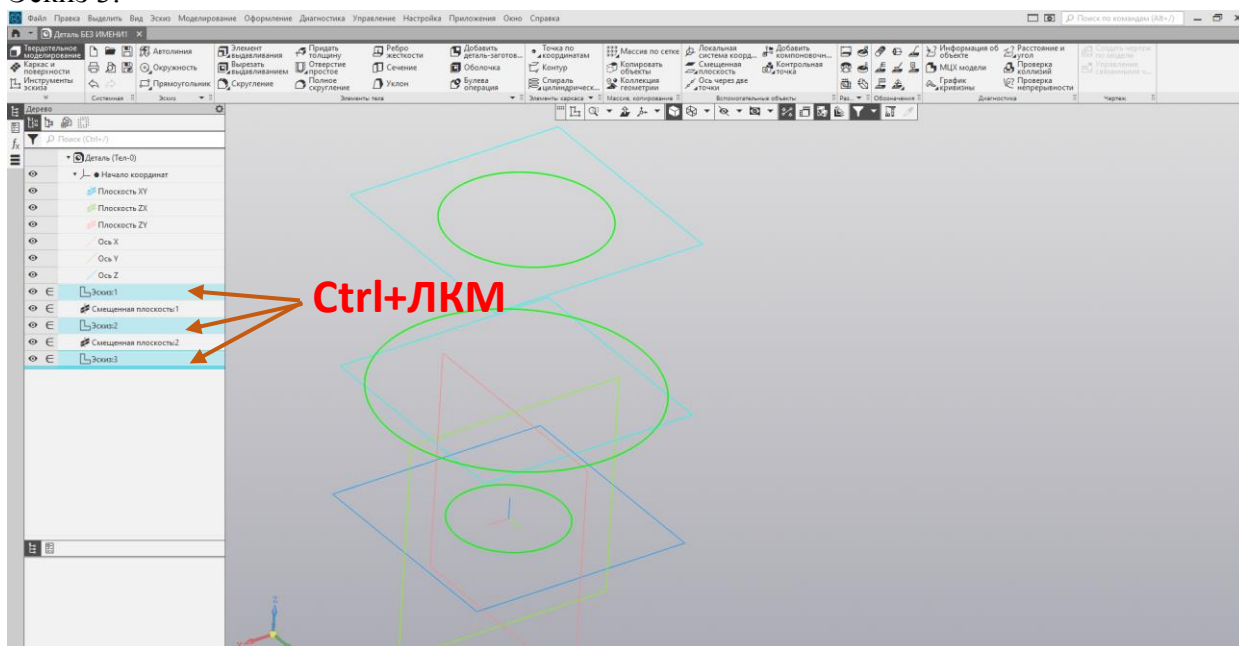
19. Ввести параметры: координаты центра - 0,0; диаметр окружности - 35 мм. Нажать кнопку **Создать**.



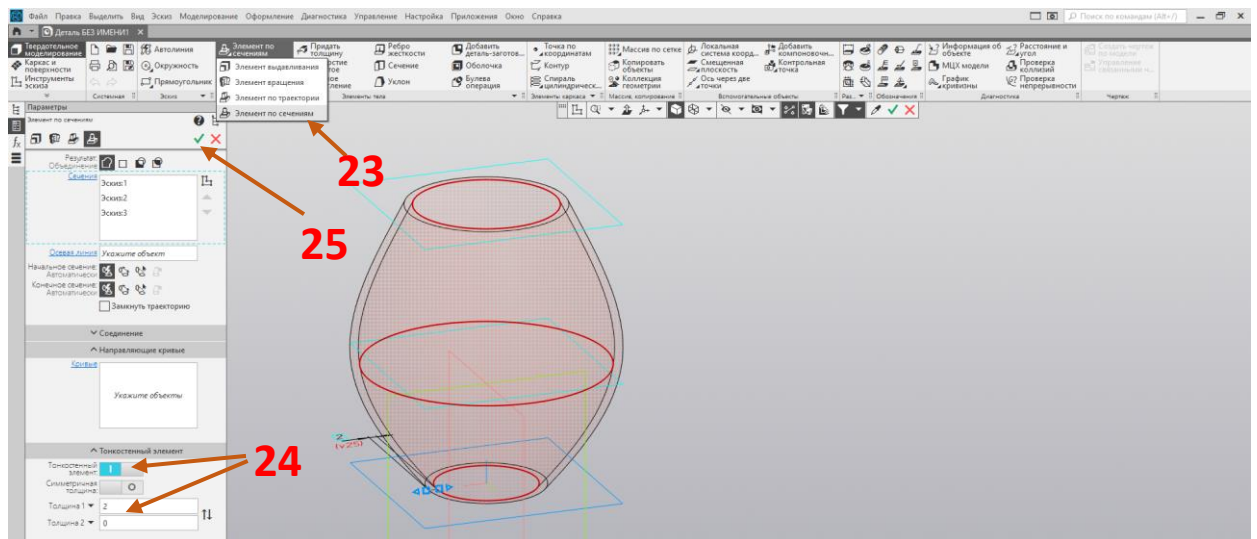
20. Закончить редактирование эскиза (повторно нажать на кнопку «эскиз»).



21. В дереве модели с зажатой клавишей Ctrl выделить поочередно Эскиз 1, Эскиз 2, Эскиз 3.

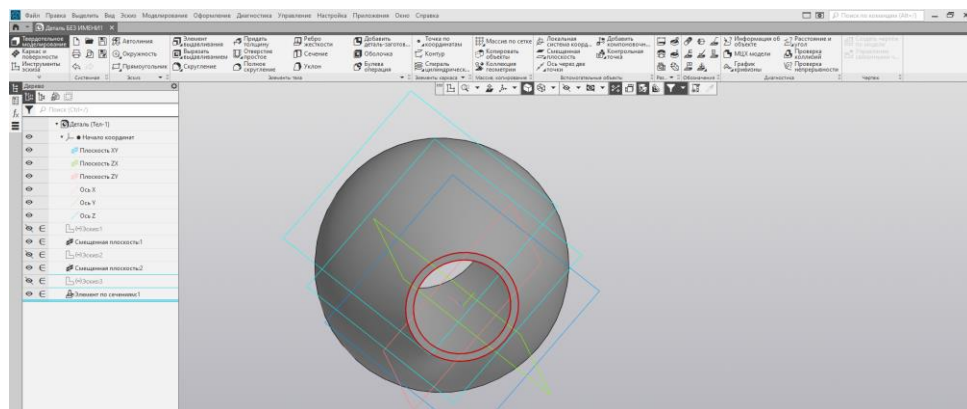


22. На панели **Элементы тела** включить инструмент **Элемент по сечениям**, в окне параметров активировать переключатель **Тонкая стенка** установить **Толщина 1 - 2 мм**. Нажать кнопку **Создать объект**.

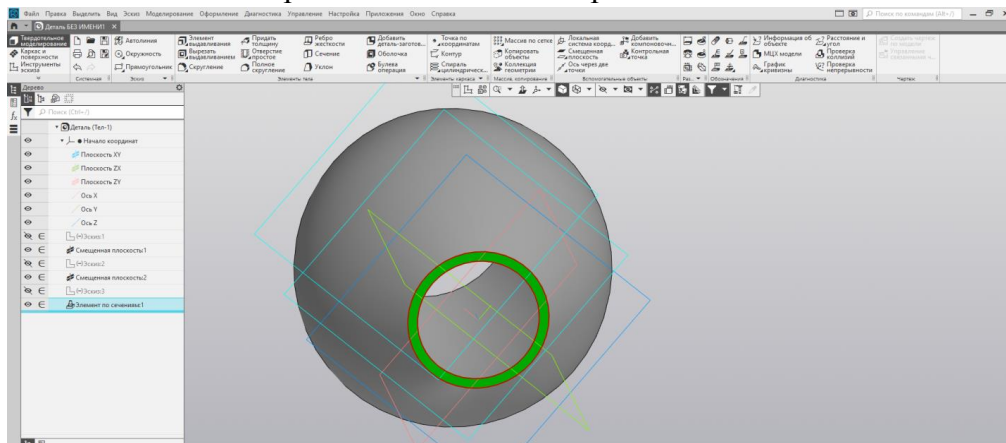


23. На экране программы должно появиться изображение модели вазы. В данном случае это только стенка вазы, дно отсутствует. Достроим дно.

24. С зажатой правой кнопкой мыши, двигая ею, сориентировать модель нижней частью корпуса к себе лицом.



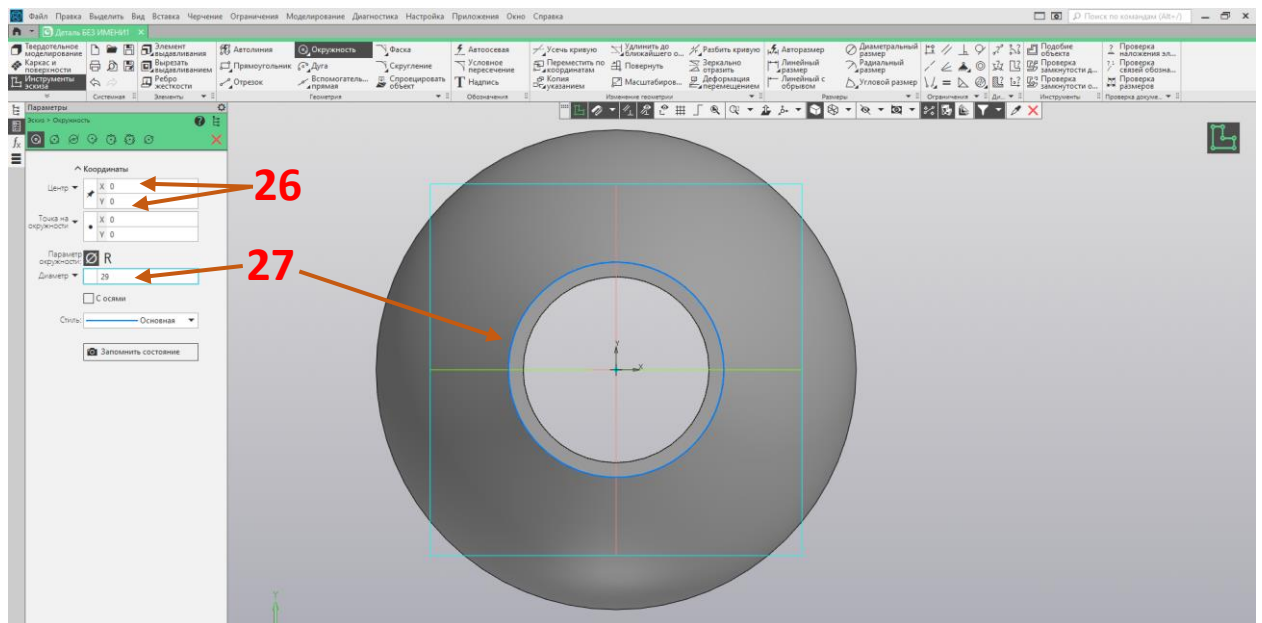
25. Выделить нижнюю грань вазы и включить режим эскиз.



26. На панели **Геометрия** выбрать инструмент **Окружность**.

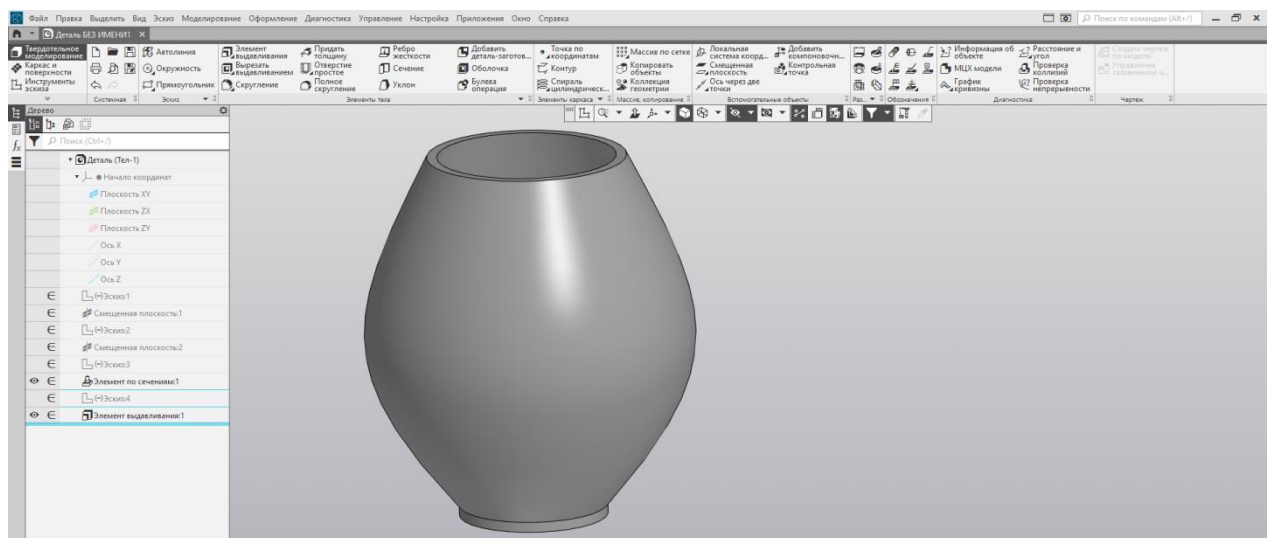
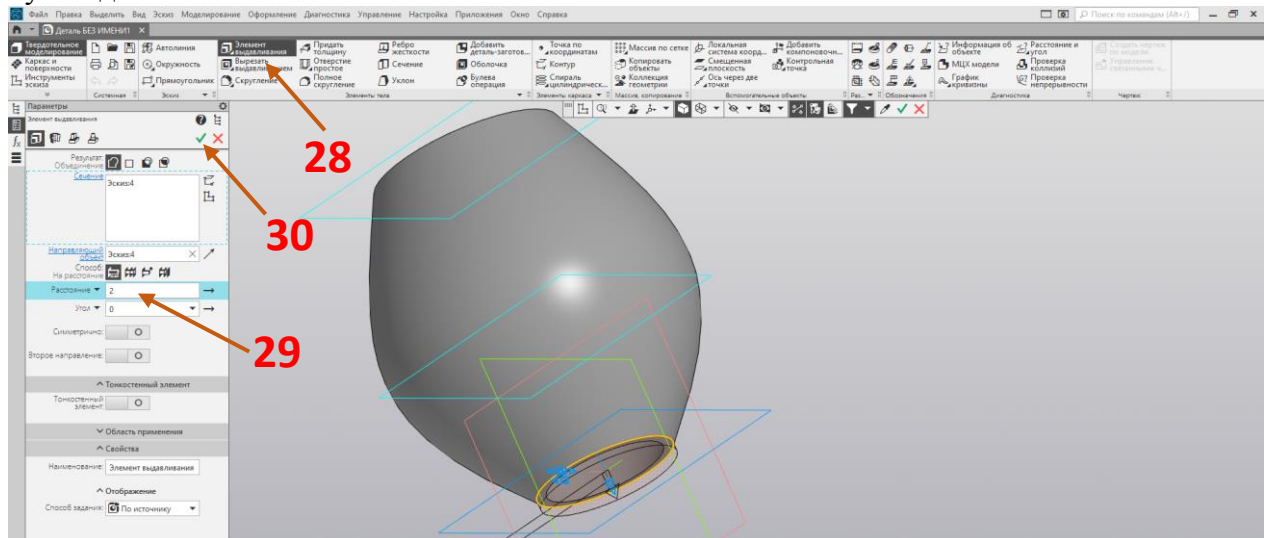
27. Ввести параметры: координаты центра - 0,0; диаметр окружности - 29 мм. Нажать кнопку **Создать**.





28. На панели **Элементы тела** выбрать **Элемент выдавливания**.

29. В окне **Параметры** в поле **Расстояние** установить значение 2 мм и нажать кнопку **Создать**.

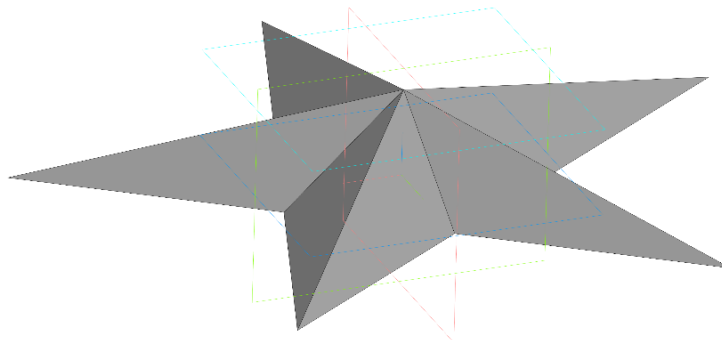


Трёхмерная модель вазы построена.

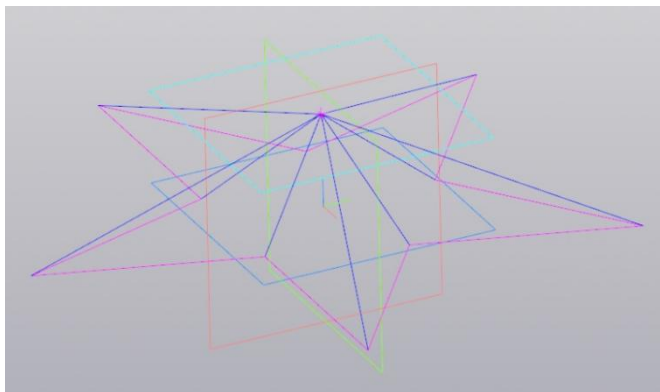
### **Задача № 9 «Пятиконечная звезда»**

**Возраст:** 12-17 лет

**Условие:** создайте 3D-модель пятиконечной звезды двумя способами, где диаметр описанной окружности основания звезды 125 мм, а ее высота 20 мм.

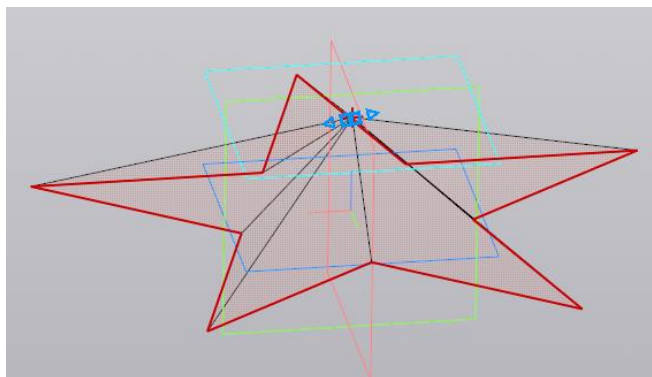


**Первый способ:**



1. Создайте смещённую плоскость.
2. На смещённой плоскости рисуйте точку.
3. На базовой плоскости нарисуйте пятиконечную звезду.
4. Произведите соединение трёхмерным отрезком углов звезды с точкой.
5. Готовый каркас сшейте с поверхностями.
6. Поверхности сшиваются и создаётся твердое тело.

**Второй способ:**

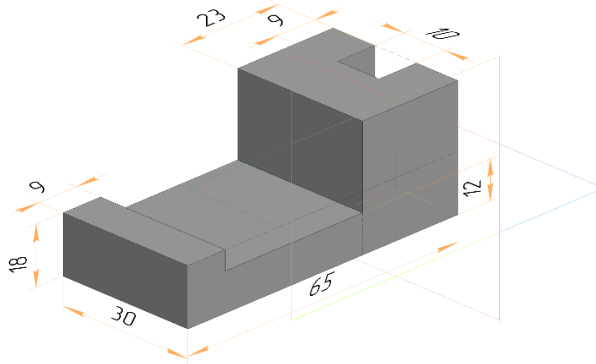


1. Создайте смещённую плоскость.
2. На смещённой плоскости нарисуйте точку.
3. На базовой плоскости нарисуйте пятиконечную звезду.
4. При помощи операции сечения соединяются точка и эскиз звезды.

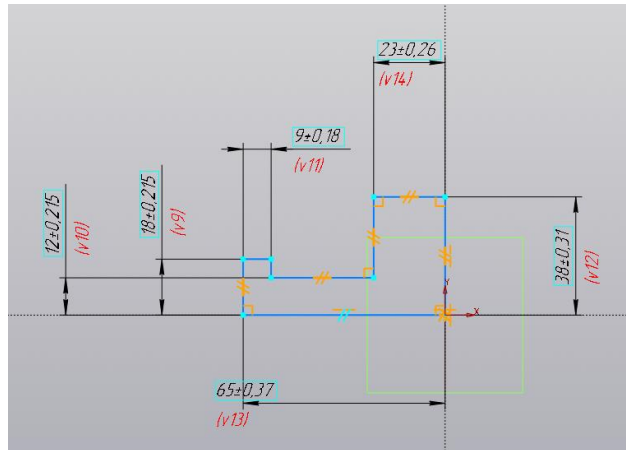
## Задача № 10 «Автолиния, авторазмер и проекция»

**Возраст:** 12-17 лет

**Условие:** создайте простую 3D-модель по размерам, используя только автолинию, ограничения, авторазмер и проекцию. Все перечисленные инструменты должны быть в обязательном порядке.



**Порядок выполнения:**



1. Создайте произвольный эскиз.
2. Установите ограничения.
3. Установите авторазмеры.
4. Эскиз выдавливается симметрично.

## Задача № 11 «3D-надпись»

**Возраст:** 12-17 лет

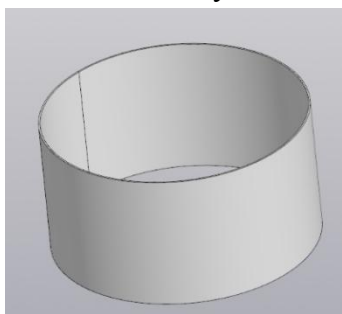
**Условие:** создайте 3D-модель с произвольными размерами поверхности с радиально расположенными буквами на ней.



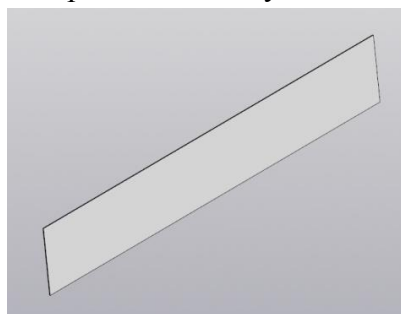


**Порядок выполнения:**

1. Создайте обечайку.



2. Разверните обечайку.



3. Выдавите надпись на плоскости.



4. Сверните обечайку.



### Задача № 12 «Гофрированная труба»

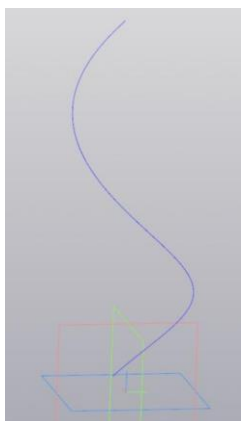
**Возраст:** 12-17 лет

**Условие:** создайте 3D-модель пустотелой гофрированной трубы с произвольными размерами.

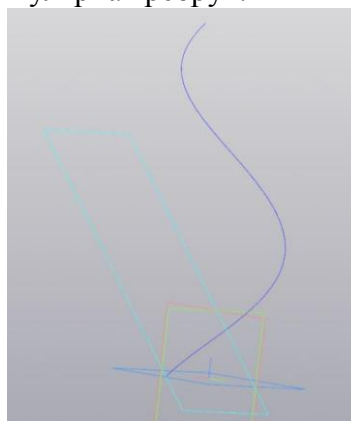


**Порядок выполнения:**

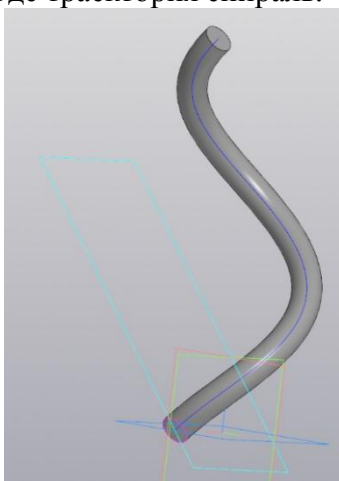
1. Нарисуйте спираль или сплайн.



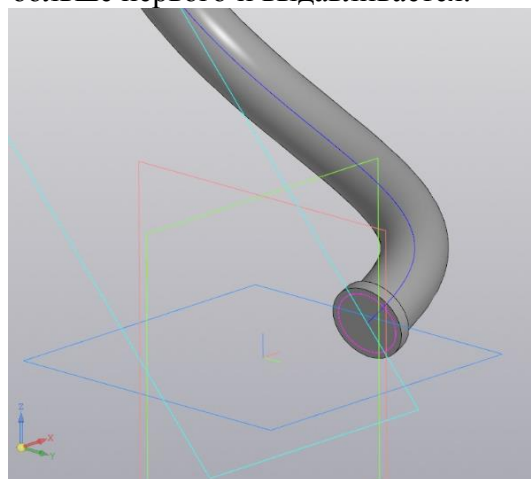
2. Установите «Плоскость через точку перпендикулярная ребру».



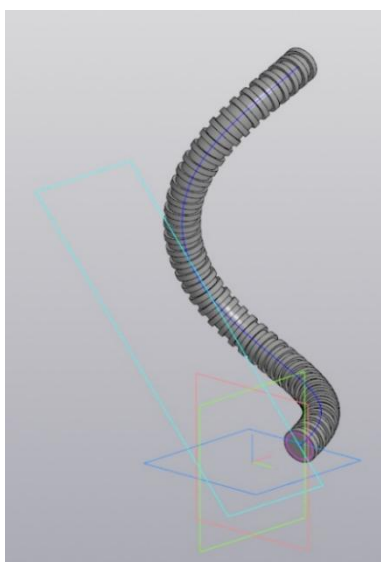
3. На плоскости нарисуйте эскиз окружности, которую выдавливают по траектории, где траектория спираль.



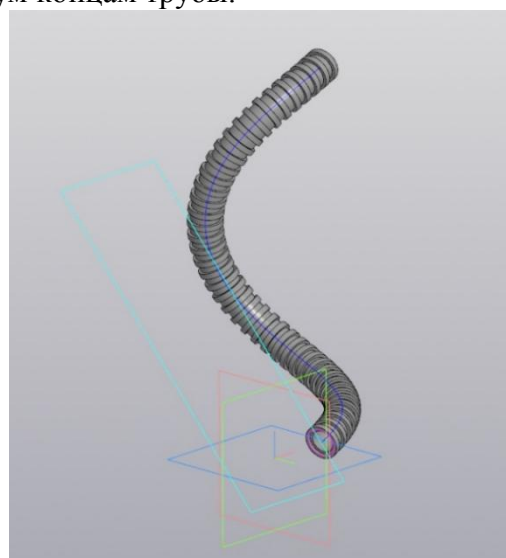
4. Нарисуйте дополнительный эскиз окружности на плоскости, который немного больше первого и выдавливается.



5. Создайте массив по траектории.



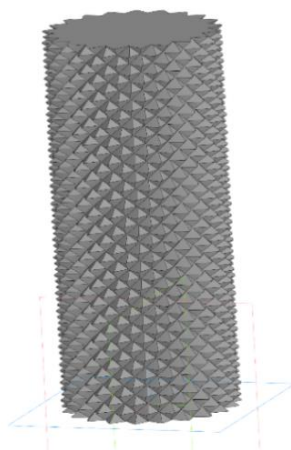
6. Примените операцию «Оболочка» по двум концам трубы.



### Задача № 13 «Рибристый цилиндр»

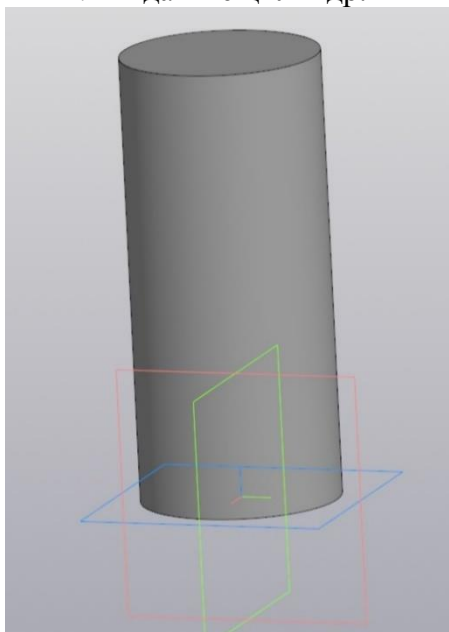
**Возраст:** 12-17 лет

**Условие:** создайте 3D-модель ребристого цилиндра с произвольными размерами.

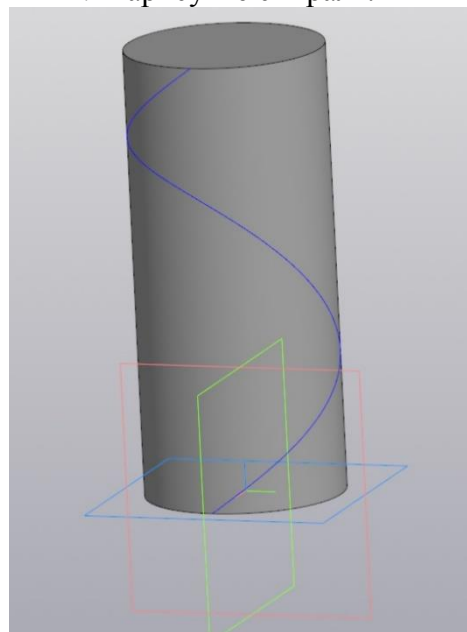


**Порядок выполнения:**

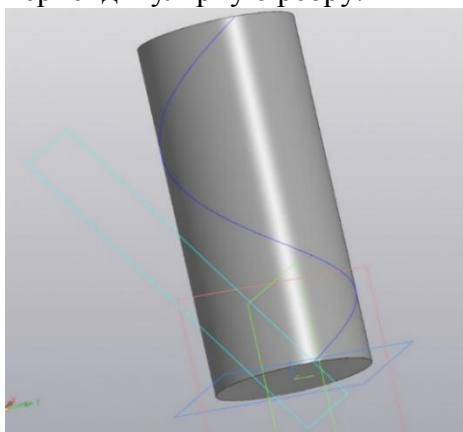
1. Выдавите цилиндр.



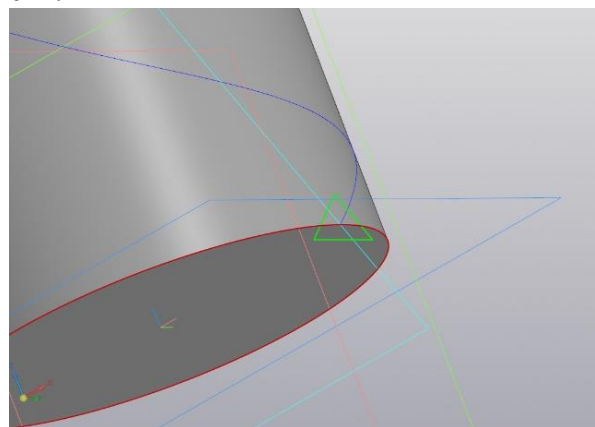
2. Нарисуйте спираль.



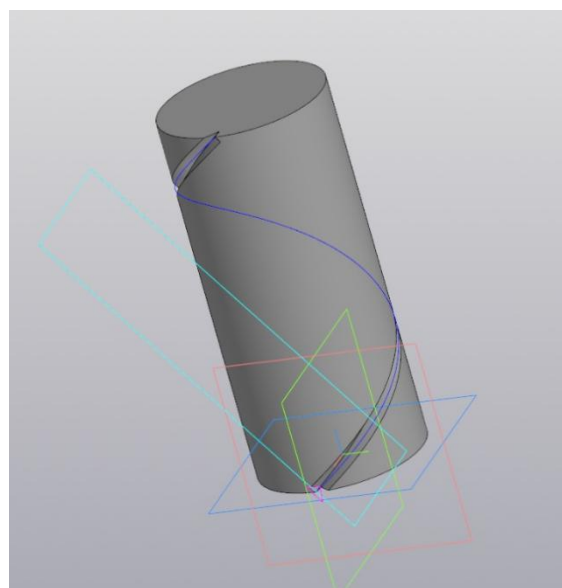
3. Установите плоскость через точку перпендикулярную ребру.



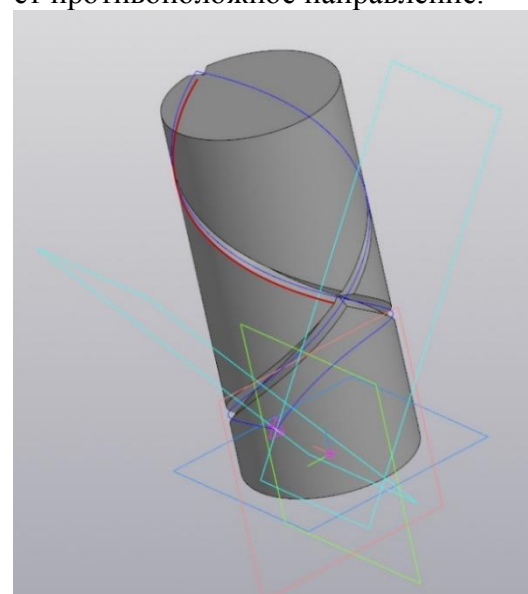
4. Нарисуйте треугольник на плоскости.



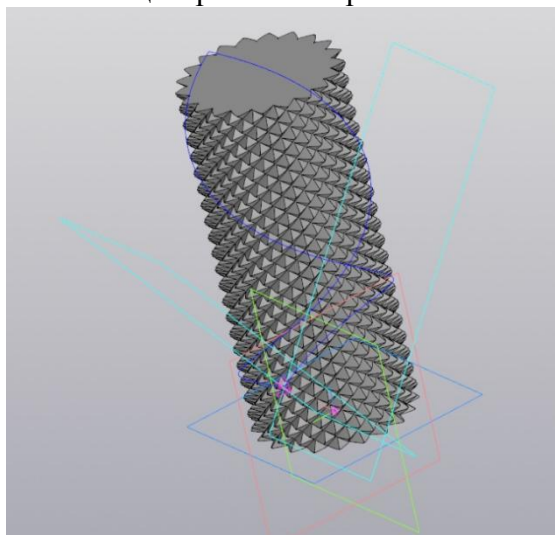
5. Вырежете треугольник по траектории.



6. Прodelайте идентичную процедуру с единственным различием, где спираль имеет противоположное направление.



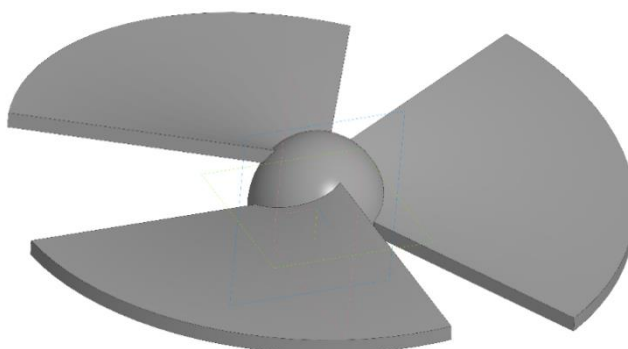
7. Примените «Массив по концентрической кривой».



### Задание № 14 «Вентилятор»

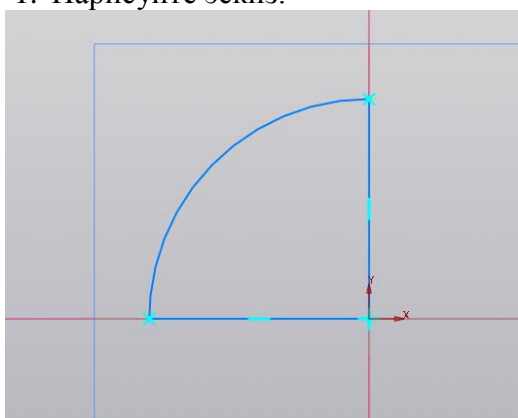
**Возраст:** 12-17 лет

**Условие:** создайте 3D-модель вентилятора с произвольными размерами.

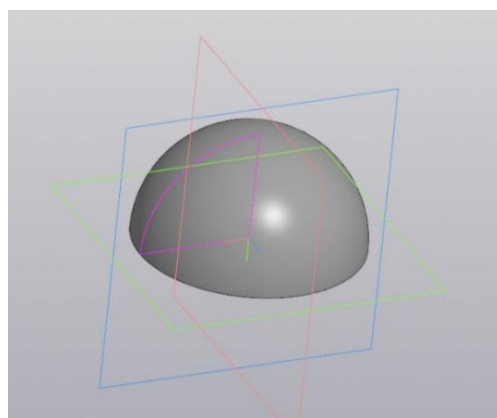


**Порядок выполнения:**

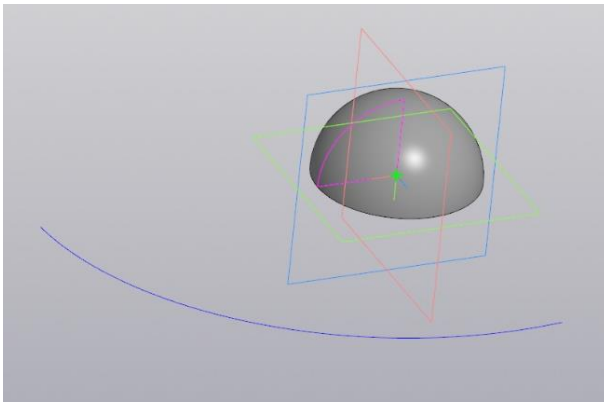
1. Нарисуйте эскиз.



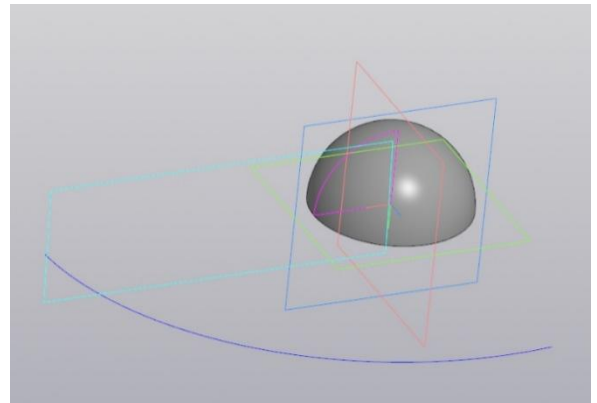
2. Нарисуйте спираль с витком 0.25.



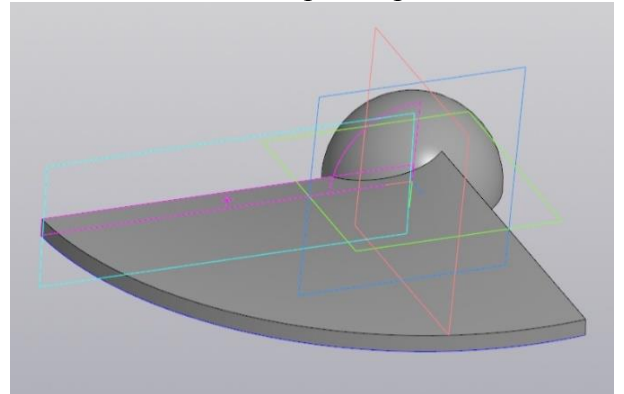
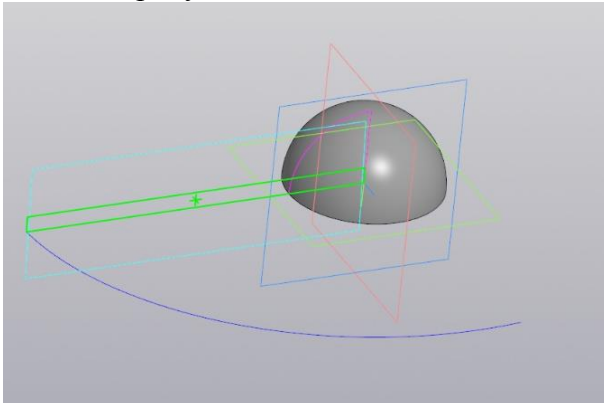
3. Установите плоскость через точку перпендикулярную кривой.



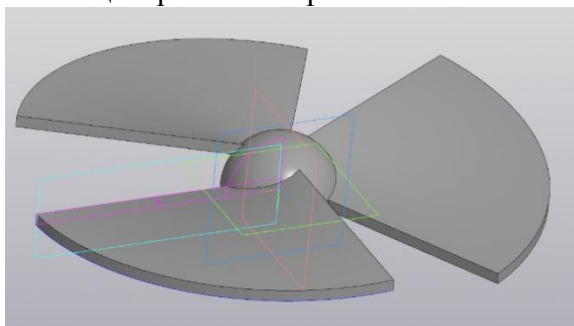
4. Нарисуйте эскиз на плоскости.



5. Выдавите по траектории.



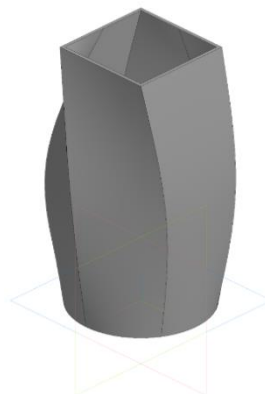
6. Сделайте массив по концентрической кривой.



## Задание № 15 «Ваза»

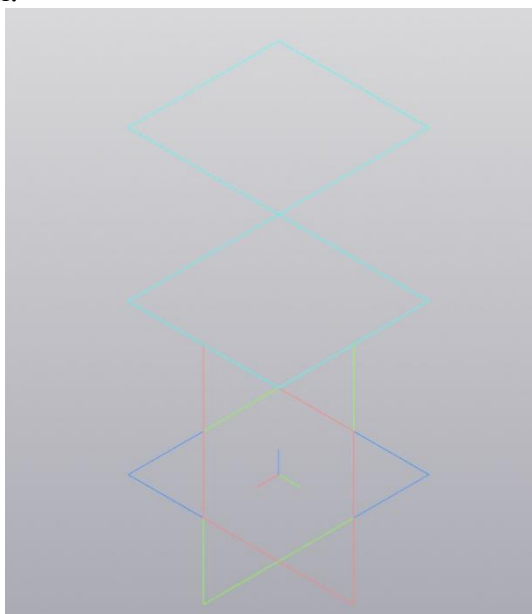
**Возраст:** 12-17 лет

**Условие:** создайте 3D-модель вазы с произвольными размерами при помощи трёх эскизов.

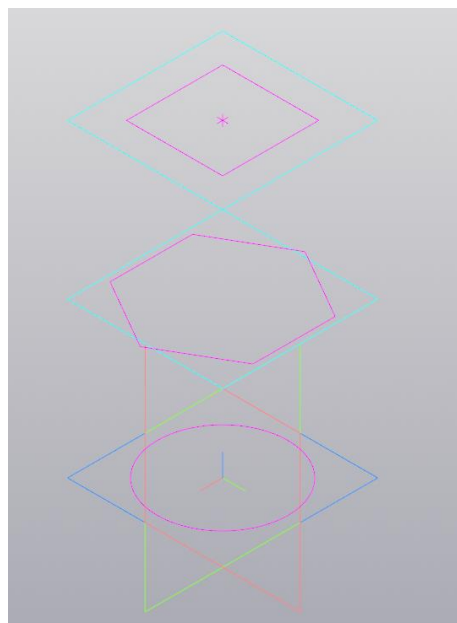


**Порядок выполнения:**

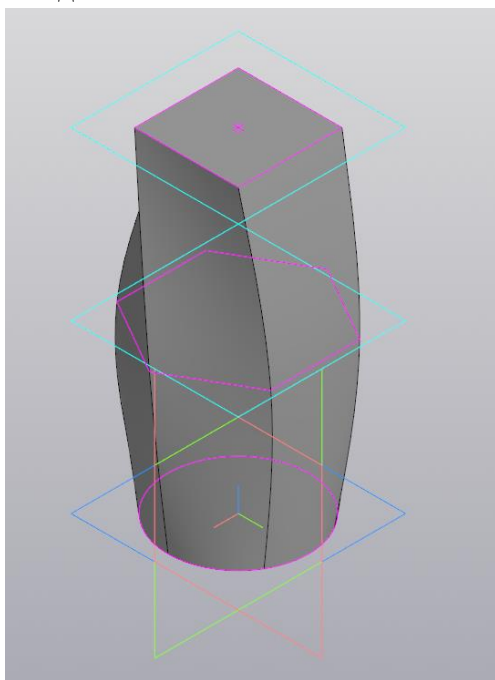
1. Нарисуйте две смещенные плоскости.



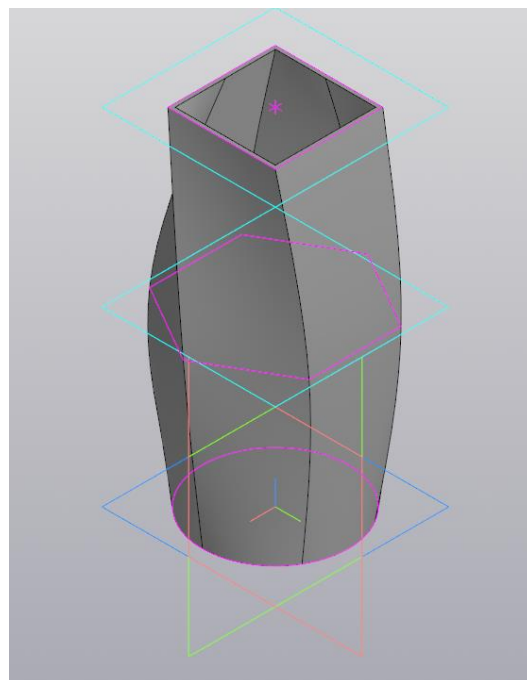
2. На плоскостях рисуется три эскиза.



3. При помощи выдавливания по сечению соединяются эскизы.



4. Сделайте оболочку.



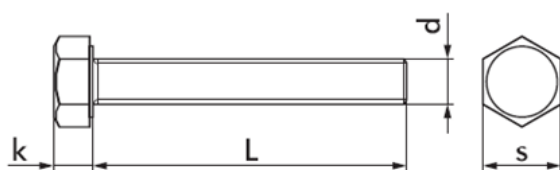
**Задача № 16 «Болт»**

**Возраст:** 12-17 лет.

**Условие:** создайте 3D-модель параметризованного болта, где условная длина тела болта (L) 100 мм.

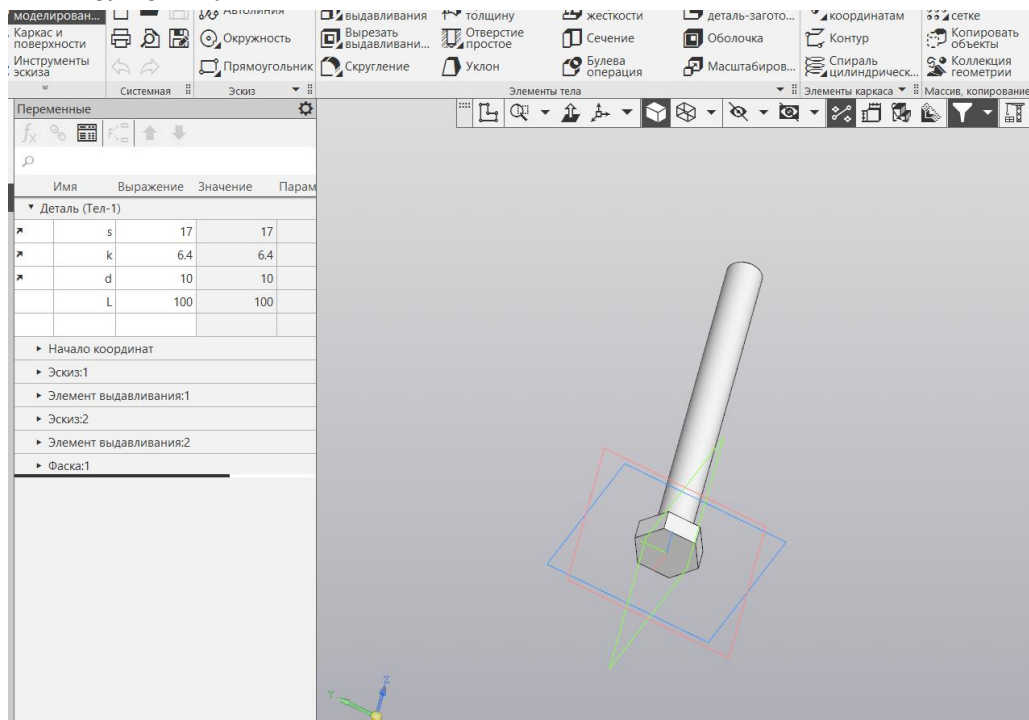


### Теоретическая масса стальных болтов с крупным шагом резьбы DIN 933, ПОЛНАЯ РЕЗЬБА



k	2.8	3.5	4	5.3	6.4	7.5	8.8	10	12.5	15	17	18,7
s	7	8	10	13	17	19	22	24	30	36	41	46
L	M4	M5	M6	M8	M10	M12	M14	M16	M20	M24	M27	M30
6	1,33	2,18	3,40									
8	1,49	2,38	3,74									
10	1,64	2,63	4,08	9,10	16,20	23,30						
12	1,80	2,87	4,42	9,80	17,20	25,00						
14	1,95	3,12	4,76	10,50	18,20	26,40						
16	2,10	3,37	5,11	11,10	19,20	27,70						
20	2,41	3,87	5,80	12,30	21,20	31,00	48,00	63,50	114,00	184,00		
25	2,80	4,49	6,65	13,90	23,70	34,10	53,00	53,00	124,00	199,00		
30	3,19	5,11	7,51	15,50	26,20	37,70	57,90	76,90	134,00	214,00		
35	3,57	5,73	8,37	17,10	28,70	41,30	62,90	83,50	145,00	229,00		
40	3,96	6,35	9,23	18,70	31,20	44,90	67,90	90,20	155,00	244,00		
45	4,34	6,99	10,10	20,30	33,70	48,50	72,90	97,10	165,00	259,00		
50	4,73	7,59	11,00	21,80	36,20	52,00	77,90	103,00	176,00	274,00		
55	5,12	8,21	11,90	23,40	38,70	55,60	82,80	110,00	186,00	289,00		
60	5,50	8,83	12,70	25,00	41,30	58,20	87,80	117,00	196,00	304,00		
65	5,89	9,45	13,60	26,60	43,80	62,80	92,80	123,00	207,00	319,00		
70	6,28	10,10	14,40	28,20	46,30	66,40	97,90	130,00	217,00	334,00	454,00	590,00
75		10,70	29,80	29,80	48,80	70,00	102,00	137,00	227,00	348,00	473,00	614,00
80		11,30	31,40	31,40	51,30	73,60	107,00	144,00	238,00	363,00	492,00	637,00
85			33,00	33,00	53,80	77,20	112,00	150,00	247,00	378,00	512,00	661,00
90			34,60	34,60	56,30	80,80	117,00	157,00	258,00	393,00	531,00	685,00
100			37,70	37,70	61,30	95,00	127,00	170,00	279,00	423,00	569,00	732,00
110			40,90	40,90	66,40		137,00	184,00	300,00	453,00	608,00	779,00
120					71,40		147,00	197,00	320,00	483,00	647,00	827,00
130					76,40		157,00	210,00	340,00	513,00	685,00	874,00
140					81,40		167,00	224,00	361,00	543,00	724,00	921,00
150					86,40		177,00	237,00	381,00	572,00	772,50	969,00
160									402,00	602,00		1010,00
180									422,00	632,00		1060,00
200									442,00	662,00		1110,00

### Порядок выполнения:



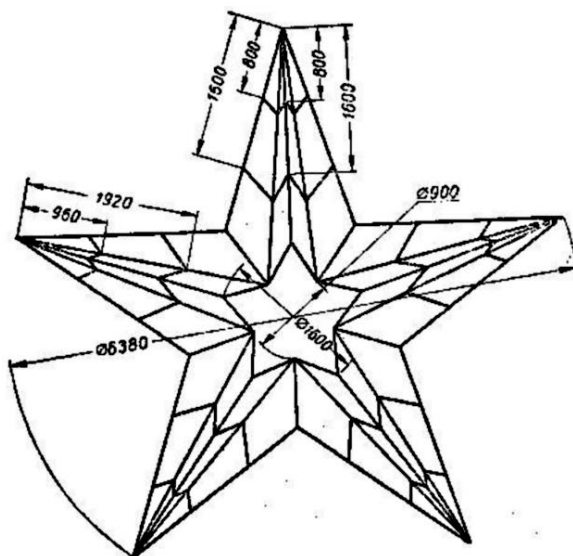
На каждом эскизе устанавливается авторазмер, где имя соответствует табличному.



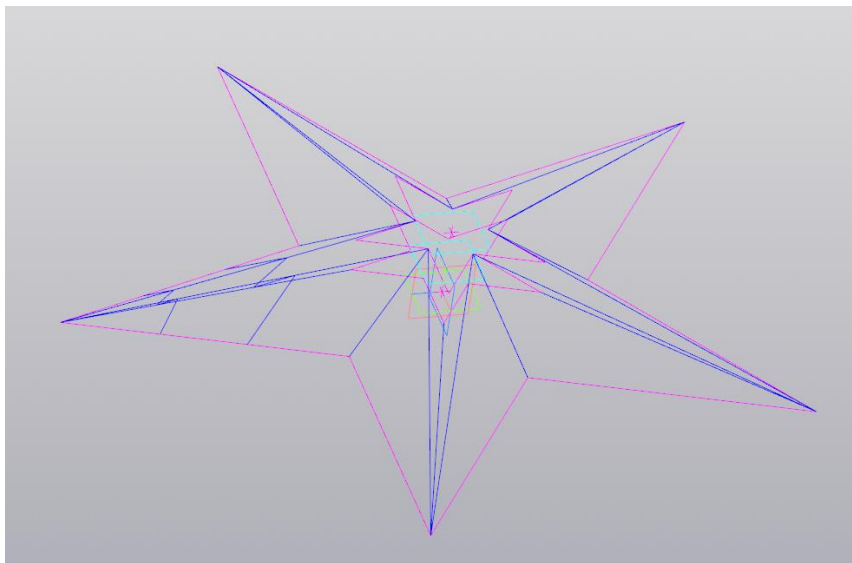
## Задание № 17 «Новогодняя звезда»

**Возраст:** 12-17 лет

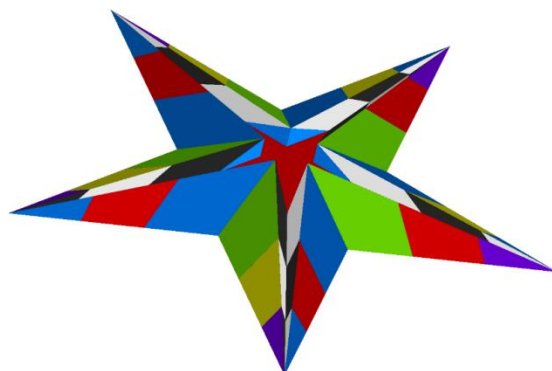
**Условие:** создайте 3D-модель «Новогодней звезды» с последующей окраской поверхности в соответствии с чертежом. Высота звезды произвольная.



**Порядок выполнения:**



1. Нарисуйте скелет звезды.
2. Обтяните скелет звезды поверхностями и сшейте.
3. В соответствии с размерами установите отрезки с последующей разбивкой поверхностей и их окраской.



### Задание № 18 «Деталь сложной конфигурации»

**Возраст:** 12-17 лет

**Условие:** создайте «приблизительную» 3D-модель детали сложной конфигурации, изображенной на рисунке.



**Порядок выполнения:**

Данная задача не имеет как такого решения. Всё упирается в фантазию и опыт человека, решающего это задание. Судейство субъективное.

## РАЗДЕЛ 3: «ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ»

*Бондарик Надежда Валерьевна,  
педагог дополнительного образования  
площадка ДТ «Кванториум-28» г. Свободный*

### Пиктопрограммирование и программирование в среде «Scratch3»

#### Задача № 1 «Пиктопрограммирование»

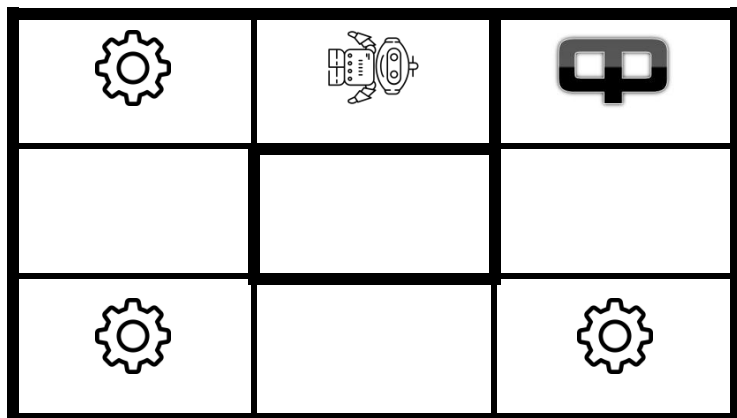
**Возраст:** 5-8 лет

Условные обозначения			
	Направление вперед		Финиш
	Поворот вправо		Нужен ремонт
	Поворот влево		Ремонт дороги



Перед вами робот-помощник, он предназначен для ремонта дорог, по которым направляются машины, при этом некоторые участки дороги повреждены и нуждаются в ремонте.

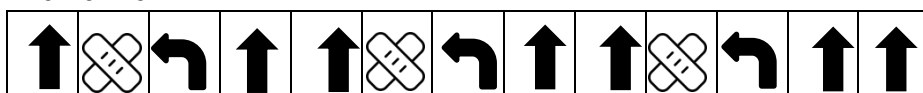
Поставьте в пустые клеточки нужные знаки, чтобы робот-помощник дошёл до финиша.



**Шаблон программы**

--	--	--	--	--	--	--	--	--	--	--	--	--	--

**Решение 1**



**Решение 2**



## Задача № 2 «Пиктопрограммирование»

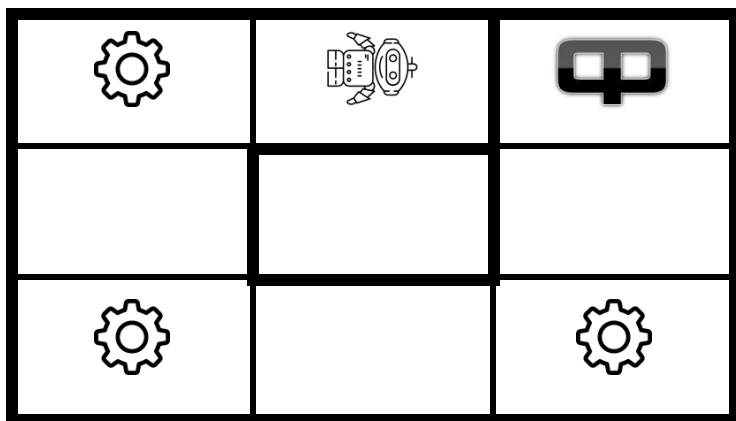
Возраст: 5-8 лет

Условные обозначения			
	Направление вперед		Финиш
	Поворот вправо		Нужен ремонт
	Поворот влево		Ремонт дороги

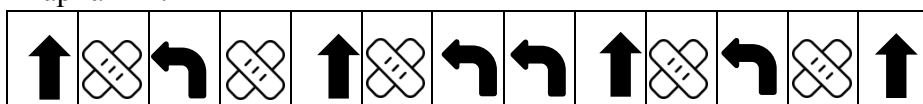


Перед вами робот-помощник, он предназначен для ремонта дорог, по которым едут машины, при этом некоторые участки дороги повреждены и нуждаются в ремонте.

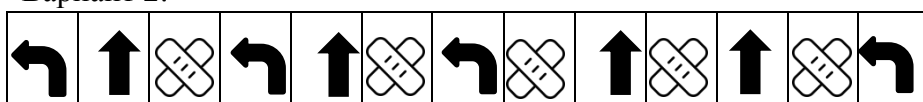
Выберите правильный вариант программы, чтобы робот-помощник дошел до финиша.



Вариант 1:



Вариант 2:



Вариант 3:



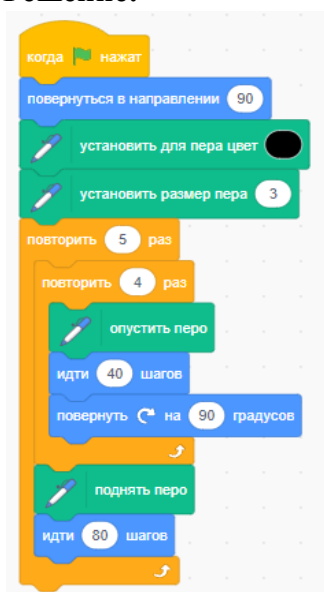
Правильный ответ: Вариант 3

### Задача № 3 «Работа с пером в Scratch3»

**Возраст:** 8-12 лет.

**Условие:** создайте программу любому спрайту, чтобы при нажатии зеленого флажка рисовались линии из пяти квадратов со стороной 60 шагов, расстояние между квадратами 20 шагов. Цвет линий может быть любой.

**Решение:**



## Задача № 4 «Олимпийские кольца»

**Возраст:** 8-12 лет

**Условие:** создайте программу пяти спрайтам, чтобы при нажатии на зелёный флажок они рисовали одновременно кольца – символ олимпийских игр.



**Решение:**

Спрайт 1	Спрайт 2	Спрайт 3	Спрайт 4	Спрайт 5
<pre> когда флажок нажат   перейти в х: -115 у: 30   повернуться в направлении 90   установить для пера цвет [синий]   установить размер пера 5   опустить перо   повторить 24 раз     идти 10 шагов     повернуть на 15 градусов         </pre>	<pre> когда флажок нажат   перейти в х: -25 у: 30   повернуться в направлении 90   установить для пера цвет [желтый]   установить размер пера 5   опустить перо   повторить 24 раз     идти 10 шагов     повернуть на 15 градусов         </pre>	<pre> когда флажок нажат   перейти в х: 70 у: 30   повернуться в направлении 90   установить для пера цвет [серый]   установить размер пера 5   опустить перо   повторить 24 раз     идти 10 шагов     повернуть на 15 градусов         </pre>	<pre> когда флажок нажат   перейти в х: -60 у: -5   повернуться в направлении 90   установить для пера цвет [зеленый]   установить размер пера 5   опустить перо   повторить 24 раз     идти 10 шагов     повернуть на 15 градусов         </pre>	<pre> когда флажок нажат   перейти в х: 30 у: -5   повернуться в направлении 90   установить для пера цвет [красный]   установить размер пера 5   опустить перо   повторить 24 раз     идти 10 шагов     повернуть на 15 градусов         </pre>

## Задача № 5 «Таймер в Scratch3»

**Возраст:** 8-12 лет

**Условие:** создайте спрайт цифра 60. Добавить костюмы (от 60 до 0). Создайте программу для таймера с переменной «Время», чтобы при изменении времени костюмы менялись в соответствии со временем. Скрыть переменную, чтобы она не отражалась на сцене. По завершении спрайт говорит о том, что время вышло.

**Решение:**

```

когда флажок нажат
  изменить костюм на Glow-9
  задать Время значение 60
  повторить 60 раз
    изменить Время на -1
    ждать 1 секунд
    следующий костюм
  говорить Время вышло! 2 секунд
  стоп все
        
```

## Программирование в среде Unity

### Задача № 1 «Перемещение противника»

**Условие:** создайте логику перемещения противника для двухмерной игры платформера в среде разработки Unity.

**Решение:**

1. Создадим класс Enemy и добавим логику для перемещения нашего противника:

```
using UnityEngine;
public class Enemy : MonoBehaviour
{
    public float walkSpeed = 5f;
    Rigidbody2D rb;
    private void Awake()
    {
        rb = GetComponent<Rigidbody2D>();
    }
    public void FixedUpdate()
    {
        rb.velocity = new Vector2 (walkSpeed * Vector2.right.x, rb.velocity.y);
    }
}
```

2. Создадим логику поведения противника, когда он подходит к стене:

```
using System;
using UnityEngine;
using static UnityEngine.Rendering.DebugUI;
[RequireComponent(typeof(Rigidbody2D), typeof(TouchingDirections))]
public class Enemy : MonoBehaviour
{
    public float walkSpeed = 5f;
    Rigidbody2D rb;
    TouchingDirections touchingDirection;
    public enum WalkableDirection { Left, Right }
    private WalkableDirection _walkDirection;
    private Vector2 walkDirectionVector = Vector2.right;
    public WalkableDirection WalkDirection
    {
        get
        {
            return _walkDirection;
        }
        set
        {
            if (_walkDirection != value)
            {

```

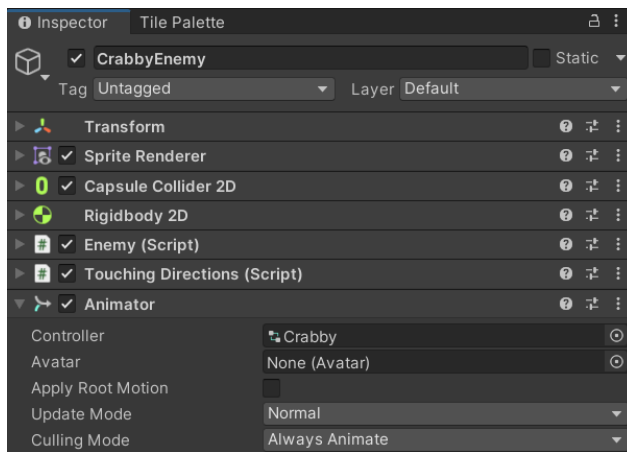
```

        gameObject.transform.localScale = new Vector2(gameObject.transform.localScale.x
* -1, gameObject.transform.lossyScale.y);
        if (value == WalkableDirection.Right)
        {
            walkDirectionVector = Vector2.right;
        }
        else if (value == WalkableDirection.Left)
        {
            walkDirectionVector = Vector2.left;
        }
    }
    _walkDirection = value;
}
}
private void Awake()
{
    rb = GetComponent<Rigidbody2D>();
    touchingDirection = GetComponent<TouchingDirections>();
}
public void FixedUpdate()
{
    if(touchingDirection.IsGrounded && touchingDirection.IsOnWall)
    {
        FlipDirection();
    }
    rb.velocity = new Vector2 (walkSpeed * walkDirectionVector.x, rb.velocity.y);//
}
private void FlipDirection()
{
    if (WalkDirection == WalkableDirection.Right)
    {
        WalkDirection = WalkableDirection.Left;
    }
    else if (WalkDirection == WalkableDirection.Left)
    {
        WalkDirection = WalkableDirection.Right;
    }
    else
    {
        Debug.Log("Недопустимое направление движения");
    }
}
}
}

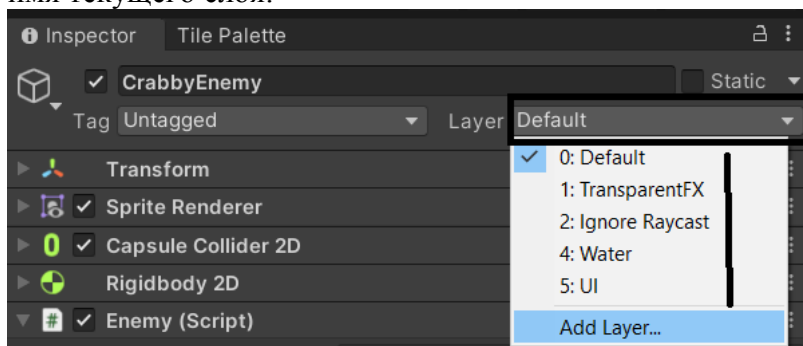
```

3. Вернемся на сцену и добавим недостающие компоненты противнику, создадим для него аниматор контроллер, добавим условия для определения границ для перемещения в аниматор:



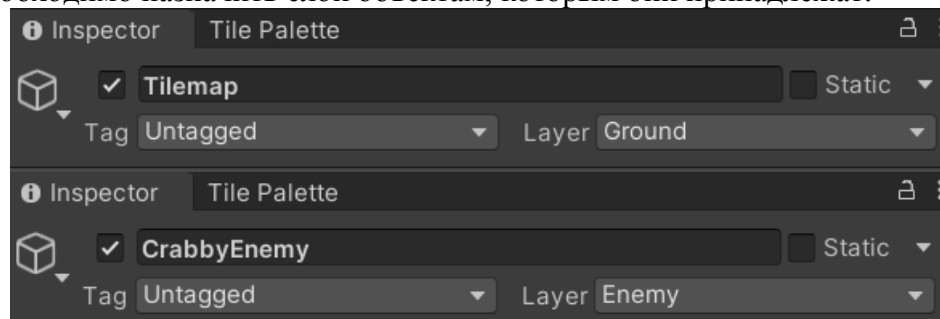


4. Сейчас противник воспринимает игрока как препятствие и разворачивается, чтобы это исправить мы можем настроить поведение противника через взаимодействие между физическими слоями, для этого добавим несколько слоев. Необходимо нажать на имя текущего слоя:

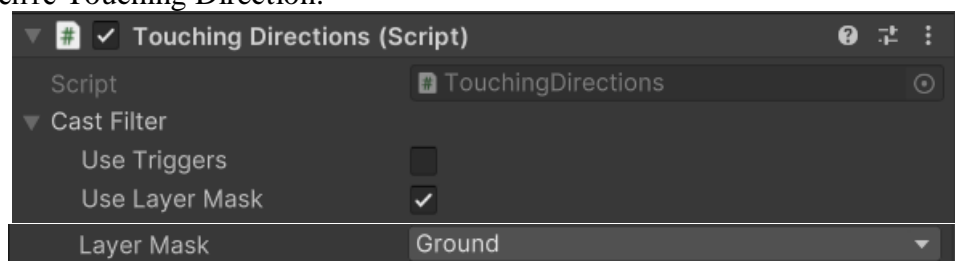


User Layer 6	Ground
User Layer 7	Player
User Layer 8	Enemy

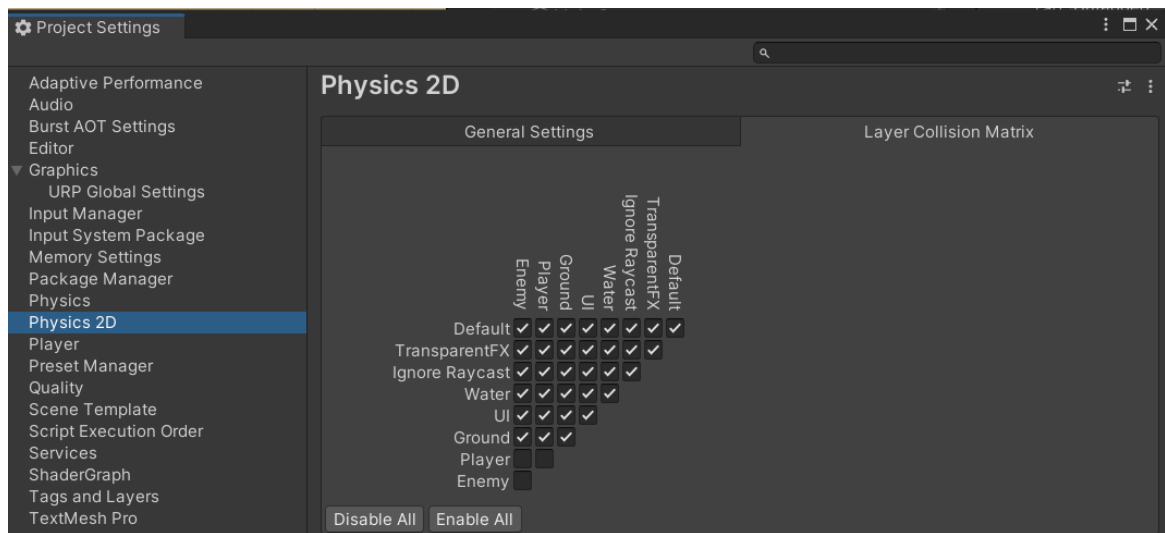
5. Необходимо назначить слои объектам, которым они принадлежат:



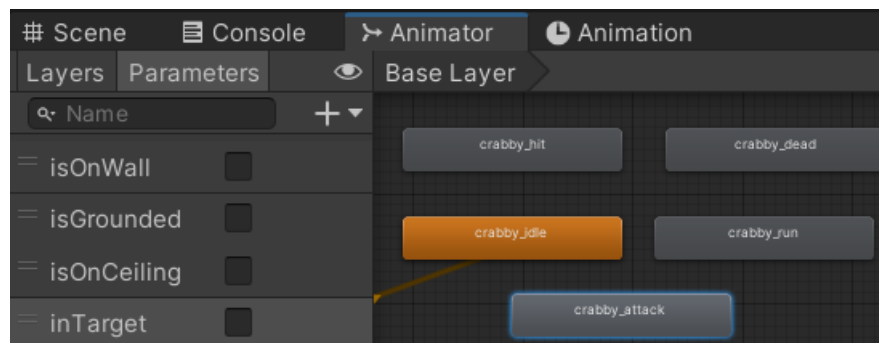
6. Также для игрока и противника необходимо назначить взаимодействие со слоями в компоненте Touching Direction:



7. Для того, чтобы противник не сталкивал персонажа, а проходил мимо мы также можем настроить пересечение слоев в настройках проекта, перейдя в основном навигационном меню Edit / Project Settings:



8. Добавим анимации для нашего противника: анимация покоя, бега, атаки, получения урона и поражения. Также добавим логическое условие для обнаружения игрока и дополнительный физический слой для зоны видимости:



Класс для обнаружения:

```
using System;
using UnityEngine;
public class TouchingDirections : MonoBehaviour
{
    public ContactFilter2D castFilter;
    public float groundDistance = 0.05f;
    public float wallDistance = 0.2f;
    public float ceilingDistance = 0.05f;

    CapsuleCollider2D touchingCollider;
    Animator animator;
    RaycastHit2D[] groundHits = new RaycastHit2D[5];
    RaycastHit2D[] wallHits = new RaycastHit2D[5];
    RaycastHit2D[] ceilingHits = new RaycastHit2D[5];

    [SerializeField] private bool _isGrounded;
    public bool IsGrounded
    {
        get
        {
            return _isGrounded;
        }
    }
}
```

```

    }
    private set
    {
        _isGrounded = value;
        animator.SetBool(AnimationStrings.isGrounded, value);
    }
}

[SerializeField] private bool _isOnWall;
public bool IsOnWall
{
    get
    {
        return _isOnWall;
    }
    private set
    {
        _isOnWall = value;
        animator.SetBool(AnimationStrings.isOnWall, value);
    }
}

[SerializeField] private bool _isOnCeiling;
private Vector2 wallCheckDirection => gameObject.transform.localScale.x > 0 ? Vector2.right : Vector2.left;
public bool IsOnCeiling
{
    get
    {
        return _isOnCeiling;
    }
    private set
    {
        _isOnCeiling = value;
        animator.SetBool(AnimationStrings.isOnCeiling, value);
    }
}

private void Awake()
{
    touchingCollider = GetComponent<CapsuleCollider2D>();
    animator = GetComponent<Animator>();
}

void FixedUpdate()
{
    IsGrounded = touchingCollider.Cast(Vector2.down, castFilter, groundHits, groundDistance) > 0;
    MiviIsOnWall = touchingCollider.Cast(wallCheckDirection, castFilter, wallHits, wallDistance) > 0;
    IsOnCeiling = touchingCollider.Cast(Vector2.up, castFilter, ceilingHits, ceilingDistance) > 0;
}
}

```

## Задача № 2 «Переход между сценами»

**Условие:** напишите класс для перехода между сценами Unity.

**Пример решения:**

### Шаг 1: Подготовка сцен

Создайте две сцены в Unity, если они еще не созданы. Для примера, назовем их Scene1 и Scene2.

Для добавления сцен в сборку проекта перейдите в File > Build Settings, нажмите Add Open Scenes, чтобы добавить текущую открытую сцену. Убедитесь, что обе сцены добавлены.

### Шаг 2: Создание скрипта перехода между сценами

В панели Project создайте новый скрипт C#. Назовите его SceneManager.

Откройте скрипт в редакторе кода по умолчанию (например, Visual Studio).

### Шаг 3: Написание кода скрипта

**Вставьте следующий код в скрипт SceneManager:**

```
using UnityEngine;
using UnityEngine.SceneManagement;

public class SceneManager : MonoBehaviour
{
    // Метод для перехода по названию сцены
    public void ChangeSceneByName(string sceneName)
    {
        SceneManager.LoadScene(sceneName);
    }
    // Метод для перехода по индексу сцены
    public void ChangeSceneByIndex(int sceneIndex)
    {
        SceneManager.LoadScene(sceneIndex);
    }
}
```

### Шаг 4: Применение скрипта

Вернитесь в Unity и создайте новый пустой объект на сцене (нажмите правую кнопку мыши > Create Empty в иерархии сцены).

Переименуйте его в SceneManager.

Перетащите скрипт SceneManager на созданный объект в инспекторе.

Для тестирования создайте UI кнопку (нажмите правую кнопку мыши в окне Hierarchy > UI > Button). Убедитесь, что для работы UI добавлен Canvas.

Выберите кнопку и в инспекторе найдите раздел On Click () в компоненте Button.

Перетащите объект SceneManager на поле, где написано None (Object), чтобы добавить его в список.

Выберите SceneManager > ChangeSceneByName или ChangeSceneByIndex и введите название или индекс сцены, на которую нужно перейти.

Теперь, нажав на кнопку в игре, вы сможете перейти на другую сцену. Не забудьте добавить сцены в сборку проекта через File > Build Settings, чтобы они были доступны для загрузки.

## Задача № 3 «Контроллер персонажа»

**Условие:** создайте контроллер персонажа для двухмерной игры в среде разработки Unity.

**Решение:**

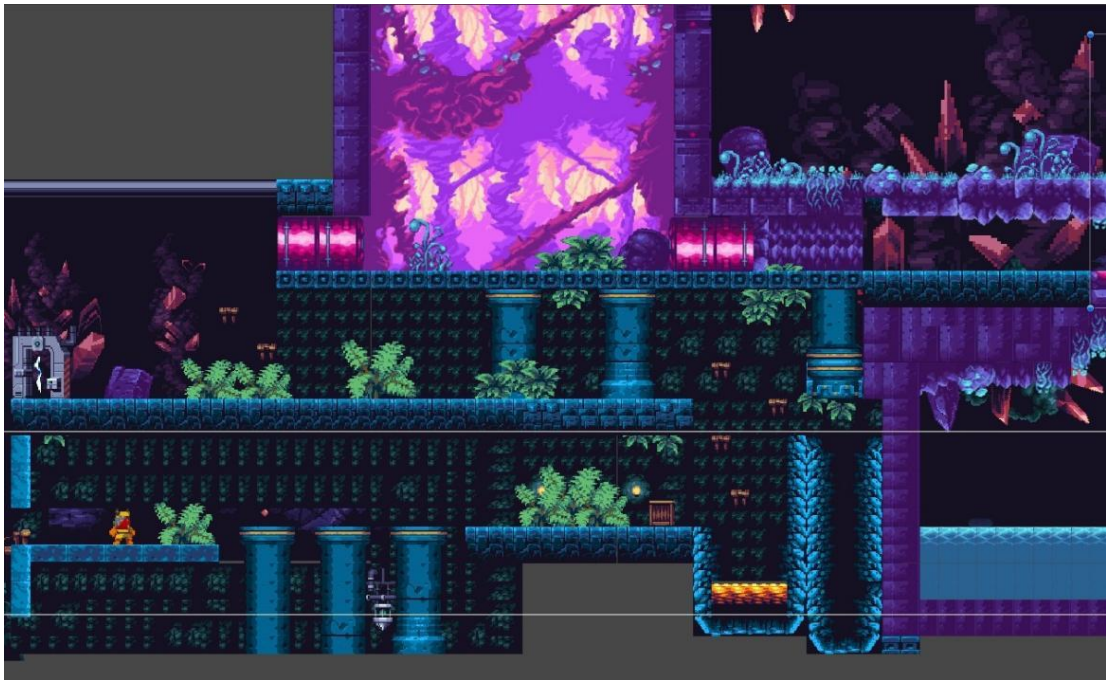
```

using UnityEngine;
public class Player : MonoBehaviour
{
    [SerializeField] private float moveSpeed;
    void Update()
    {
        float horizontal = Input.GetAxis("Horizontal");
        float vertical = Input.GetAxis("Vertical");
        Vector3 step = new Vector3(horizontal, vertical, 0);
        transform.position += step * (moveSpeed * Time.deltaTime);
    }
}

```

**Задача № 4 «Двухмерный уровень»**

**Условие:** создайте двухмерный уровень для игры платформера в среде разработки Unity. С помощью встроенных инструментов создайте двухмерный начальный уровень, продолжительностью не менее 10 экранов камеры. Уровень должен содержать платформы разных высот элементы окружения, начальную точку и конечную для перехода на следующий уровень. Задача уровня познакомить игрока с базовыми игровыми механиками такими как прыжок, перемещение и взаимодействие с элементами окружения.

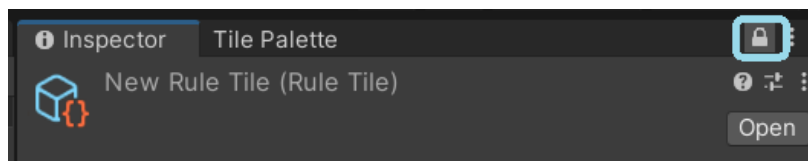
**Решение:****Рекомендации:**

Быстрое создание уровней с помощью Tile Rule.

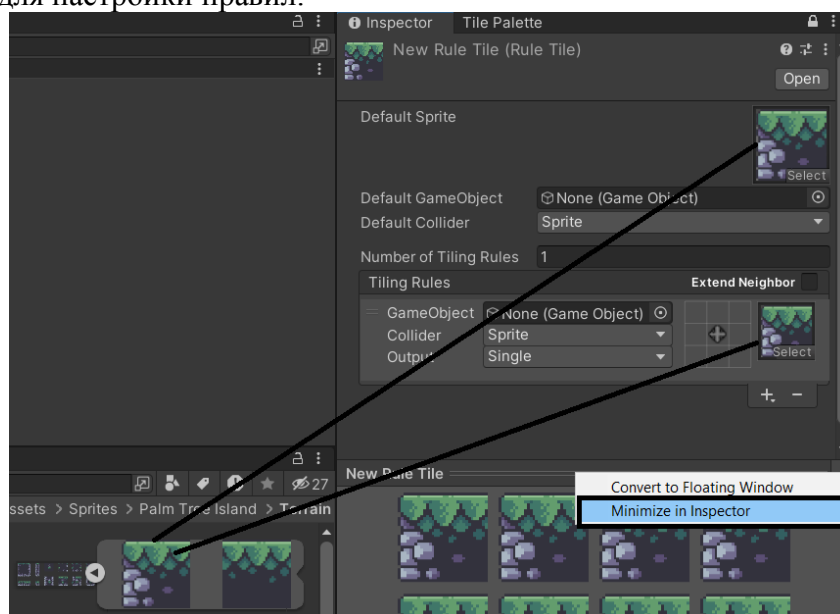
Для начала нам нужно добавить файл, который отвечает за правила:

2D	>	Sprites	>	Tile
Visual Scripting	>	Tiles	>	Animated Tile
Shader Graph	>	Tile Palette	>	Rule Tile
Shader	>	Sprite Shape Profile	>	Isometric Rule Tile
Shader Variant Collection	>	Physics Material 2D	>	Hexagonal Rule Tile
Testing	>	Sprite Atlas	>	Rule Override Tile

Для удобства добавления следующих элементов зафиксируем, вкладку правил в инспекторе:



Далее нужно настроить правила для каждого спрайта, из которых мы будем создавать уровень. Окно предпросмотра правил можно открыть, нажав на 3 точки внизу и выбрав **Minimize in Inspector**. Теперь нужно установить спрайт по умолчанию и добавить новые спрайты для настройки правил.

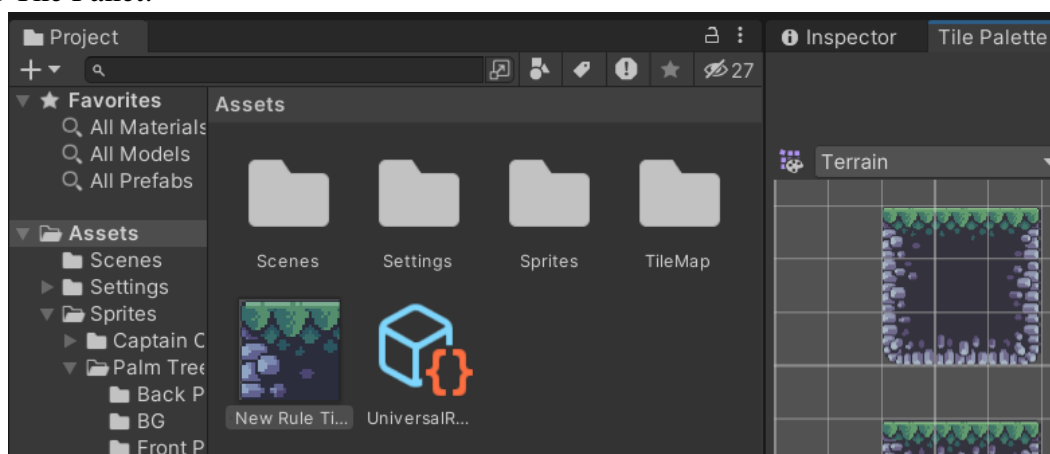


После того как вы настроили все правила. Нужно перетащить файл с правилами в Tile Pallet.

Если перетащить весь файл со спрайтами правила автоматически откроются для всех блоков.



Для дальнейшего строительства используйте блок с правилами. Его также нужно добавить в Tile Pallet.





## Задача № 5 «Источник света»

**Условие:** разработайте скрипт, который будет включать и выключать источник света в сцене, по нажатию клавиши в Unity.

### Пример решения:

1. Добавьте источник света:  
Разместите на сцене Directional Light или любой другой источник света.
2. Создание скрипта:  
Создайте новый скрипт C# и назовите его LightController.  
Добавьте следующий код:  
using UnityEngine;  
public class LightController : MonoBehaviour  
{  
    public Light myLight; // Переменная для источника света  
  
    void Update()  
    {  
        if(Input.GetKeyDown(KeyCode.L)) // Проверяем нажатие клавиши L  
        {  
            myLight.enabled = !myLight.enabled; // Включаем или выключаем свет  
        }  
    }  
}

### Применение скрипта:

Примените скрипт к любому объекту на сцене.  
В инспекторе укажите источник света, который будет контролироваться скриптом.

## Задача № 6 «Движущаяся платформа»

**Условие:** создайте движущуюся платформу в Unity.

### Пример решения:

#### Шаг 1: Подготовка сцены

На сцене создайте объект, который будет выступать в роли движущейся платформы.  
Создайте пустые объекты (GameObject > Create Empty), которые будут представлять точки, между которыми будет двигаться платформа. Расположите эти объекты в пространстве сцены в нужных местах.

#### Шаг 2: Создание скрипта для движения платформы

В панели Project создайте новый скрипт C#. Назовите его MovingPlatform.  
Откройте скрипт в редакторе кода по умолчанию.

#### Шаг 3: Написание кода скрипта

**Вставьте следующий код в скрипт MovingPlatform:**

```
using UnityEngine;  
public class MovingPlatform : MonoBehaviour  
{  
    public Transform[] points; // Массив точек, между которыми будет перемещаться платформа  
    public float speed = 2f; // Скорость движения платформы  
    private int currentIndex = 0; // Текущий индекс точки, к которой движется платформа  
  
    void Update()  
    {
```

```

        if (points.Length == 0) return; // Если точек нет, выходим из метода

        Transform targetPoint = points[currentIndex]; // Получаем текущую целевую точку
        transform.position = Vector3.MoveTowards(transform.position, targetPoint.position,
        speed * Time.deltaTime); // Перемещаем платформу к целевой точке

        // Проверяем, достигла ли платформа целевой точки
        if (transform.position == targetPoint.position)
        {
            currentIndex++; // Переходим к следующей точке
            if (currentIndex >= points.Length) // Если достигли последней точки, начинаем снача
                ла
            {
                currentIndex = 0;
            }
        }
    }
}

```

#### Шаг 4: Применение скрипта

Вернитесь в Unity и перетащите скрипт MovingPlatform на объект платформы в инспекторе.

Выберите платформу и в инспекторе найдите поле Points в скрипте MovingPlatform. Укажите количество точек и присвойте каждому элементу массива соответствующий пустой объект сцены, который вы создали ранее как точку.

Теперь при запуске сцены платформа будет перемещаться между заданными точками. Вы можете регулировать скорость перемещения платформы, изменяя значение Speed в инспекторе.

## Задача № 7 «Параллакс эффект»

**Условие:** создайте параллакс эффект для двухмерной игры платформера в среде разработки Unity.

Параллакс – это эффект, который создаёт иллюзию глубины. Например, когда вы едете в поезде и смотрите в окно, столбы проносятся мимо вас быстро, деревья и дома – со средней скоростью, а горы на фоне почти статичны. Вот эта разная скорость перемещения объектов на разной глубине и есть параллакс.

#### Решение:

1. Создание параллакс эффекта для заднего фона, для этого нам нужно сделать несколько слоев для фона. Создадим дополнительный класс ParallaxEffect `using UnityEngine;`

```

public class ParallaxEffect : MonoBehaviour
{
    public Camera cam;
    public Transform followTarget;

    Vector2 startingPosition;
    float startingZ;
    Vector2 camMoveSinceStart => (Vector2)cam.transform.position - startingPosition;
    float zDistanceFromTarget => transform.position.z - followTarget.transform.position.z;
}

```

```

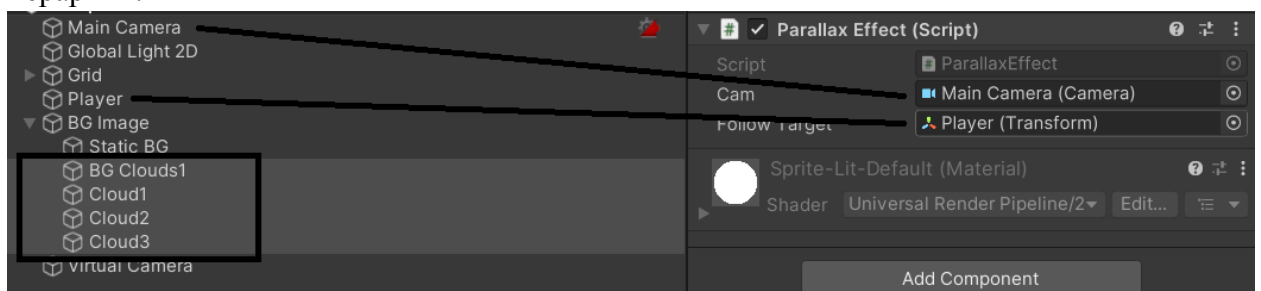
float clippingPlane => (cam.transform.position.z + (zDistanceFromTarget > 0 ?
cam.farClipPlane : cam.nearClipPlane));
float parallaxFactor => Mathf.Abs(zDistanceFromTarget) / clippingPlane;

void Start()
{
    startingPosition = transform.position;
    startingZ = transform.position.z;
}

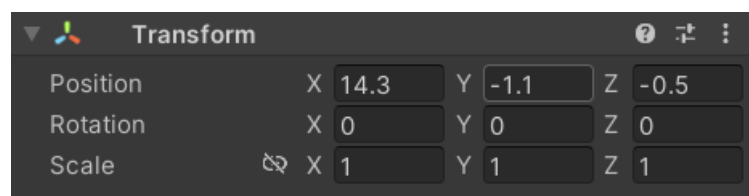
void Update()
{
    Vector2 newPosition = startingPosition + camMoveSinceStart * parallaxFactor;
    transform.position = new Vector3(newPosition.x, newPosition.y, startingZ);
}
}

```

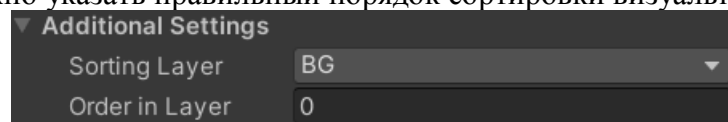
2. Теперь настроим данный эффект на сцене. Необходимо добавить компонент ParallaxEffect для объектов, которые будут задействованы в эффекте, и настроить компонент. Можно настроить несколько объектов одного типа предварительно выбрав их в окне иерархии.



3. Нужно установить разные координаты по оси z. Чем дальше объект будет находиться по данной позиции, тем быстрее он будет перемещаться по отношению к игроку.



4. Также нужно указать правильный порядок сортировки визуальных слоев



## Задача № 8 «Звуковая дорожка»

**Условие:** разработайте систему в Unity, которая позволяет добавлять несколько звуковых дорожек в проект и переключаться между ними по нажатию клавиши или кнопки на интерфейсе пользователя.

**Пример решения:**

### Шаг 1: Подготовка аудиофайлов

Подготовьте несколько аудиофайлов, которые хотите использовать в проекте. Это могут быть музыкальные треки, звуковые эффекты или любые другие аудиозаписи.

Импортируйте аудиофайлы в Unity, перетащив их в панель Project.

### **Шаг 2: Создание объекта AudioSource**

Создайте на сцене новый пустой объект (GameObject > Create Empty) и назовите его, например, AudioManager.

Добавьте к AudioManager компонент AudioSource через инспектор (Компоненты > Audio > AudioSource).

### **Шаг 3: Настройка AudioSource**

Отключите галочку Play On Awake в компоненте AudioSource, чтобы трек не начинал играть автоматически при старте сцены.

Установите галочку Loop, если хотите, чтобы трек за цикливался.

### **Шаг 4: Создание скрипта для переключения аудио**

Создайте новый скрипт C# и назовите его AudioSwitcher.

Откройте скрипт и добавьте следующий код:

```
using UnityEngine;
public class AudioSwitcher : MonoBehaviour
{
    public AudioClip[] tracks; // Массив аудиотреков
    private AudioSource audioSource; // Источник аудио
    private int currentTrackIndex = 0; // Индекс текущего трека
    private void Start()
    {
        audioSource = GetComponent<AudioSource>();
        PlayCurrentTrack();
    }
    private void Update()
    {
        // Переключение треков по нажатию клавиши T (или любой другой)
        if (Input.GetKeyDown(KeyCode.T))
        {
            currentTrackIndex = (currentTrackIndex + 1) % tracks.Length;
            PlayCurrentTrack();
        }
    }
    void PlayCurrentTrack()
    {
        audioSource.clip = tracks[currentTrackIndex];
        audioSource.Play();
    }
}
```

Примените скрипт к объекту AudioManager.

### **Шаг 5: Настройка скрипта в Unity**

В инспекторе для объекта AudioManager укажите количество элементов в массиве Tracks и добавьте ваши аудиотреки перетаскиванием их из панели Project в каждый элемент массива.

Теперь, запустив сцену и нажимая клавишу T (или любую другую, указанную в коде), вы сможете переключаться между заданными звуковыми дорожками.

## Программирование на языке Python и C#

### Задача № 1 «Работа с вводом/выводом»

**Возраст:** 12-14 лет

**Условие:** напишите программу, которая считает сумму двух чисел, введенных пользователем.

**Решение:**

```
# Программа для вычисления суммы двух чисел
num1 = float(input("Введите первое число: "))
num2 = float(input("Введите второе число: "))
result = num1 + num2
print("Сумма равна:", result)
```

### Задача № 2 «Цикл for»

**Возраст:** 12-14 лет

**Условие:** создайте программу, которая выводит таблицу умножения для определенного числа (например,  $5 \times 1 = 5$ ,  $5 \times 2 = 10$  и т. д.)

**Решение:**

```
# Программа для вывода таблицы умножения
number = int(input("Введите число для таблицы умножения: "))
for i in range(1, number + 1):
    print("%d x %d = %d" % (number, i, number * i))
```

### Задача № 3 «Условный оператор»

**Возраст:** 12-14 лет

**Условие:** напишите программу, которая определяет, является ли число чётным или нечётным.

**Решение:**

```
# Программа для проверки на четность
number = int(input("Введите число для проверки на четность: "))
if number % 2 == 0:
    print("Число", number, "четно.")
else:
    print("Число", number, "нечетно.")
```

### Задача № 4 «Количество гласных букв»

**Возраст:** 12-14 лет

**Условие:** создайте программу, которая подсчитывает количество гласных букв во введенном пользователем слове.

**Решение:**

```
# Программа для подсчета гласных букв
word = input("Введите слово для подсчета гласных букв: ")
vowels = ["a", "y", "o", "и", "э", "ы", "я", "ю", "е", "ё"]
count = sum([letter in vowels for letter in word])
print("Количество гласных букв в этом слове:", count)
```

### Задача № 5 «Проверка на простое число»

**Возраст:** 12-14 лет

**Условие:** напишите программу, которая определяет, является ли введенное число простым.

**Решение:**

```
# Программа для проверки на простое число
number = int(input("Введите число для проверки на простоту: "))
isPrime = True
for i in range(2, number // 2 + 1):
    if number % i == 0:
        isPrime = False
        break
if isPrime:
    print(number, "это простое число.")
else:
    print(number, "не является простым числом.")
```

### Задача № 6 «Перевод температуры»

**Возраст:** 12-14 лет

**Условие:** разработайте программу для конвертации температуры из градусов Цельсия в градусы Фаренгейта.

**Решение:**

```
# Программа для конвертации температур
tempCelcius = float(input("Введите температуру в градусах Цельсия: "))
tempFahrenheit = (tempCelcius * 9/5) + 32
print("Температура в градусах Фаренгейта:", tempFahrenheit)
```

### Задача № 7 «Генерация случайных чисел»

**Возраст:** 12-14 лет

**Условие:** создайте программу, которая генерирует случайное число и предлагает пользователю угадать его.

**Решение:**

```
import random
# Генерация случайного числа
randomNumber = random.randint(1, 100)
guess = None
while guess != randomNumber:
    guess = int(input("Угадайте число от 1 до 100: "))
    if guess < randomNumber:
        print("Число меньше, попробуйте ещё раз.")
```



```
elif guess > randomNumber:  
    print("Число больше, попробуйте ещё раз.")  
print("Вы угадали! Число было:", randomNumber)
```

### Задача № 8 «Определение возрастной категории»

**Возраст:** 12-14 лет

**Условие:** создайте программу, которая по возрасту определяет категорию: ребенок, подросток, взрослый.

**Решение:**

```
# Программа для определения возрастной категории  
age = int(input("Введите ваш возраст: "))  
if age < 13:  
    print("Ребенок")  
elif age < 20:  
    print("Подросток")  
else:  
    print("Взрослый")
```

*Хмелёв Алексей Павлович,  
педагог дополнительного образования  
мобильного ДТ «Кванториум-28»*

### Задача № 9 «Число и делители»

**Возраст:** 13-15 лет

**Условие:** напишите программу на языке C# или Python, которая будет принимать от пользователя число, и будет выводить на экран все его делители.

**Материально-техническое обеспечение:** ноутбук с установленным IDE для Python и/или C#

Примеры:

Ввод: 27      Вывод: 1 3 9 27

Ввод: 100      Вывод: 1 2 4 5 10 20 25 50 100

**Решение:**

```
number = int(input("Введите число: "))  
  
print("Делители числа", number, ":")  
  
for i in range(1, number + 1):  
  
    if number % i == 0:  
  
        print(i)
```

### Задача № 10 «Арифметические действия»

**Возраст:** 13-15 лет

**Условие:** напишите программу на языке C# или Python, которая будет принимать от пользователя два числа и оператор (+, -, \*, /) и будет выводить на экран результат действия оператора на эти числа.

**Материально-техническое обеспечение:** ноутбук с установленным IDE для Python и/или C#

Примеры:

Ввод: 2 + 5      Вывод: 7

Ввод: 3 \* 10      Вывод: 30

**Решение:**

```
num1 = float(input("Введите первое число: "))
num2 = float(input("Введите второе число: "))
operator = input("Введите оператор (+, -, *, /): ")
if operator == '+':
    result = num1 + num2
    print(f"{num1} + {num2} = {result}")
elif operator == '-':
    result = num1 - num2
    print(f"{num1} - {num2} = {result}")
elif operator == '*':
    result = num1 * num2
    print(f"{num1} * {num2} = {result}")
elif operator == '/':
    if num2 == 0:
        print("Ошибка: деление на ноль!")
    else:
        result = num1 / num2
        print(f"{num1} / {num2} = {result}")
else:
    print("Некорректный оператор. Пожалуйста, используйте только '+', '-', '*', или '/'.")
```

## Задача № 11 «Простые числа»

**Возраст:** 13-15 лет

**Условие:** напишите программу на языке C# или Python, которая будет принимать от пользователя число N и будет выводить на экран первые N простых чисел.

**Материально-техническое обеспечение:** ноутбук с установленным IDE для Python и/или C#

Примеры:

Ввод: 5      Вывод: 2 3 5 7 11

Ввод: 7      Вывод: 2 3 5 7 11 13 17

**Решение:**

```
def is_prime(num):  
    if num < 2:  
        return False  
    for i in range(2, int(num**0.5) + 1):  
        if num % i == 0:  
            return False  
    return True  
  
def first_n_primes(n):  
    prime_numbers = []  
    num = 2  
    while len(prime_numbers) < n:  
        if is_prime(num):  
            prime_numbers.append(num)  
        num += 1  
    return prime_numbers  
  
N = int(input("Введите число N: "))  
if N < 1:  
    print("Пожалуйста, введите число больше 0.")  
else:  
    prime_list = first_n_primes(N)  
    print(f"Первые {N} простых чисел:")  
    for prime in prime_list:  
        print(prime)
```

**Задача № 12 «Квадратное уравнение»****Возраст:** 13-15 лет**Условие:** напишите программу на языке C# или Python, которая будет принимать от пользователя числа a, b, c и будет выводить на экран корни уравнения  $ax^2+bx+c=0$ **Материально-техническое обеспечение:** ноутбук с установленным IDE для Python и/или C#

Примеры:

Ввод: 1 2 1	Вывод: -1
Ввод: 1 -5 6	Вывод: 2 3
Ввод: 1 1 1	Вывод: корней нет

**Решение:**

```

import math

def solve_quadratic_equation(a, b, c):
    discriminant = b**2 - 4*a*c

    if discriminant > 0:
        x1 = (-b + math.sqrt(discriminant)) / (2*a)
        x2 = (-b - math.sqrt(discriminant)) / (2*a)
        print(f"У уравнения {a}x^2 + {b}x + {c} = 0 есть два корня:")
        print(f"x1 = {x1}")
        print(f"x2 = {x2}")
    elif discriminant == 0:
        x = -b / (2*a)
        print("У уравнения есть один корень:")
        print(f"x = {x}")
    else:
        print("У уравнения нет действительных корней.")

# Запрос пользователю ввода коэффициентов a, b, c
a = float(input("Введите коэффициент a: "))
b = float(input("Введите коэффициент b: "))
c = float(input("Введите коэффициент c: "))
solve_quadratic_equation(a, b, c)

```

### Задача № 13 «Палиндром»

**Возраст:** 13-15 лет

**Условие:** Напишите программу на языке C# или Python, которая будет проверять, является ли введённая строка палиндромом.

**Материально-техническое обеспечение:** ноутбук с установленным IDE для Python и/или C#

Примеры:

Ввод: искатьтакси	Вывод: Палиндром
Ввод: машина	Вывод: Не палиндром

**Решение:**

```

def is_palindrome(s):
    # Удаляем из строки пробелы и приводим к нижнему регистру
    s = s.replace(" ", "").lower()

    # Сравниваем строку с перевернутой строкой

```

```

    return s == s[::-1]

# Запрос пользователю ввода строки
user_input = input("Введите строку для проверки на палиндром: ")
if is_palindrome(user_input):
    print("Введенная строка является палиндромом.")
else:
    print("Введенная строка не является палиндромом.")

```

### Задача № 14 «Среднее арифметическое»

**Возраст:** 13-15 лет

**Условие:** Напишите программу на языке C# или Python, которая будет выводить на экран среднее арифметическое всех чисел, введенных пользователем (программа перестаёт считать числа, когда пользователь вводит -1)

**Материально-техническое обеспечение:** ноутбук с установленным IDE для Python и/или C#

Примеры:

Ввод: 1 2 3 4 5 6 -1	Вывод: 3.5
Ввод: 2 7 1 1 -1	Вывод: 2.75

**Решение:**

```

numbers = []

while True:
    user_input = input("Введите число (или -1 для завершения): ")
    if user_input == "-1":
        break
    try:
        number = float(user_input)
        numbers.append(number)
    except ValueError:
        print("Пожалуйста, введите корректное число.")

if len(numbers) == 0:
    print("Вы не ввели ни одного числа.")
else:
    average = sum(numbers) / len(numbers)
    print(f"Среднее арифметическое введенных чисел: {average}")

```

## **Задача № 15 «Текстовый редактор»**

**Возраст:** 14-17 лет

**Условие:** реализуйте текстовый редактор на языке программирования C# с использованием Windows Forms или WPF. Редактор должен позволять пользователю вводить текст, сохранять его в файл и открывать файлы для редактирования.

**Материально-техническое обеспечение:** ноутбук, программный комплекс «Visual Studio» версии от 2015 года.

**Ход выполнения:**

1. Создание интерфейса:

Используйте Windows Forms или WPF для создания интерфейса текстового редактора. Добавьте текстовое поле для ввода текста и кнопки для сохранения и открытия файлов.

2. Обработка текста:

Напишите код на C#, который будет обрабатывать ввод и вывод текста в файл. При нажатии на кнопку «Сохранить», текст должен сохраняться в файле, а при открытии файла - отображаться в текстовом поле.

3. Добавление функционала:

Добавьте функционал для форматирования текста и другие дополнительные функции по желанию. Например, вы можете добавить возможность выбора шрифта и размера текста, выполнение поиска и замены текста и т. д.

**Решение:**

4. Создание формы.
5. Добавить текстовое поле.
6. Добавить кнопку сохранить.
7. Написание кода.

**Примерный код для формы:**

```
// Файл Form1.cs
using System;
using System.IO;
using System.Windows.Forms;
namespace TextEditorApp
{
    public partial class Form1 : Form
    {
        public Form1()
        {
            InitializeComponent();
        }
        // Обработчик события нажатия кнопки "Сохранить"
        private void SaveButton_Click(object sender, EventArgs e)
        {
            // Создание диалогового окна для сохранения файла
            SaveFileDialog saveFileDialog = new SaveFileDialog();
            saveFileDialog.Filter = "Text files (*.txt)|*.txt|All files (*.*)|*.*";
            // Если пользователь выбрал файл и нажал "ОК"
```



```

        if (saveFileDialog.ShowDialog() == DialogResult.OK)
        {
            // Запись текста из текстового поля в выбранный файл
            File.WriteAllText(saveFileDialog.FileName, textBox1.Text);
        }
    }
    // Обработчик события нажатия кнопки "Открыть"
    private void OpenButton_Click(object sender, EventArgs e)
    {
        // Создание диалогового окна для выбора файла для открытия
        OpenFileDialog openFileDialog = new OpenFileDialog();
        openFileDialog.Filter = "Text files (*.txt)|*.txt|All files (*.*)|*.*";
        // Если пользователь выбрал файл и нажал "ОК"
        if (openFileDialog.ShowDialog() == DialogResult.OK)
        {
            // Чтение текста из выбранного файла и вывод его в текстовое поле
            textBox1.Text = File.ReadAllText(openFileDialog.FileName);
        }
    }
}

```

### Задача № 16 «Дополни код»

**Возраст:** 12-17 лет

**Условие:** в следующем коде реализован метод для вычисления среднего значения чисел из массива. Однако в нем есть пропущенные слова и строки кода. Ваша задача - правильно дополнить код так, чтобы метод работал корректно.

**Материально-техническое обеспечение:** ноутбук, программный комплекс «Visual Studio» версии от 2015 года.

```

using System;
public class Program
{
    public static void Main()
    {
        int[] numbers = { 5, 10, 15, 20, 25 };
        double average = CalculateAverage(_____);
        Console.WriteLine("Среднее значение: " + average);
    }
    public static double CalculateAverage(_____)
    {
        int sum = 0;
        foreach (int num in numbers)
        {
            sum += num;
        }
        double average = sum / numbers._____;
        return _____;
    }
}

```

**Требуемый результат:**

После запуска программы на консоль должно выводиться среднее значение чисел из массива numbers.

**Решение:**

```

using System;
public class Program
{
    public static void Main()
    {
        int[] numbers = { 5, 10, 15, 20, 25 };
        double average = CalculateAverage(numbers);
        Console.WriteLine("Среднее значение: " + average);
    }
    public static double CalculateAverage(int[] numbers)
    {
        int sum = 0;
        foreach (int num in numbers)
        {
            sum += num;
        }
        double average = sum / numbers.Length;
        return average;
    }
}

```

**Задача № 17 «Калькулятор»****Возраст:** 14-17 лет**Условие:** написать простой консольный калькулятор со следующими арифметическими действиями: +,-,/,\*. Также проверить, выдаёт ли ошибку при делении на «0»**Материально-техническое обеспечение:** ноутбук, программный комплекс «Visual Studio» версии от 2015 года**Решение:****Пример на языке C#**

```

using System;
class Calculator
{
    static void Main(string[] args)
    {
        Console.WriteLine("Добро пожаловать в калькулятор!");
        while (true)
        {
            Console.WriteLine("Введите первое число:");
            double num1 = Convert.ToDouble(Console.ReadLine());
            Console.WriteLine("Введите операцию (+, -, *, /):");
            string operation = Console.ReadLine();
            Console.WriteLine("Введите второе число:");
            double num2 = Convert.ToDouble(Console.ReadLine());
            double result = 0;
            // Выполнение операции в зависимости от введенного символа операции
            switch (operation)
            {
                case "+":
                    result = num1 + num2;
                    break;
                case "-":

```

```

        result = num1 - num2;
        break;
    case "*":
        result = num1 * num2;
        break;
    case "/":
        // Проверка деления на ноль
        if (num2 != 0)
        {
            result = num1 / num2;
        }
        else
        {
            Console.WriteLine("Ошибка: нельзя делить на ноль!");
            continue; // Продолжить цикл снова, не выводя результат
        }
        break;
    default:
        Console.WriteLine("Ошибка: некорректная операция!");
        continue; // Продолжить цикл снова, не выводя результат
    }
    Console.WriteLine("Результат: " + result);
    // Проверка, хочет ли пользователь продолжить вычисления
    Console.WriteLine("Хотите продолжить вычисления? (да/нет):");
    string choice = Console.ReadLine();
    if (choice.ToLower() != "да")
    {
        break; // Выход из цикла, если пользователь не хочет продолжать
    }
}
Console.WriteLine("До свидания!");
}
}

```

## Задача № 18 «Отгадай число»

**Возраст:** 14-17 лет

**Условие:** создайте консольную игру, в которой компьютер загадывает случайное число от 0 до 100, а пользователь должен попытаться угадать его за 10 попыток. После каждой попытки пользователю выводится подсказка о том, насколько он близок к загаданному числу.

**Материально-техническое обеспечение:** ноутбук, программный комплекс «Visual Studio» версии от 2015 года.

**Правила:**

1. Пользователь вводит целое число от 0 до 100.
2. После каждого ввода отображается подсказка, сообщающая, насколько близко к загаданному числу находится введенное число.
3. Пользователь имеет 10 попыток для угадывания числа.
4. Если пользователь угадал число, игра завершается. Если нет, выводится сообщение о проигрыше.

**Решение:**

**Пример на языке программирования C#**

```
using System;
```

```

class GuessNumberGame
{
    static void Main(string[] args)
    {
        Random random = new Random();
        int secretNumber = random.Next(0, 101);
        int attempts = 10;
        Console.WriteLine("Компьютер загадал число от 0 до 100. У вас есть 10 попыток, что-
бы угадать.");
        for (int i = 1; i <= attempts; i++)
        {
            Console.Write($"Попытка {i}: Введите число от 0 до 100: ");
            int guess = Convert.ToInt32(Console.ReadLine());
            int difference = Math.Abs(secretNumber - guess);
            if (guess == secretNumber)
            {
                Console.WriteLine($"Точно! Вы угадали число {secretNumber}!");
                return; // Завершение программы
            }
            else if (difference <= 10)
            {
                Console.WriteLine($"Тепло ({difference})");
            }
            else
            {
                Console.WriteLine($"Холодно ({difference})");
            }
        }
        Console.WriteLine($"К сожалению, вы исчерпали все попытки. Загаданное число было
{secretNumber}.");
    }
}

```

### Задача № 19 «Больше, меньше или равно?»

**Возраст:** 12-17 лет

**Условие:** необходимо написать консольную программу, где пользователь будет вводить с клавиатуры 2 числа. Числа будут сравниваться с последующим выводом в консоль результата этого сравнения (равны ли значения, а если нет, то какое число больше/меньше).

**Материально-техническое обеспечение:** ноутбук, программный комплекс «Visual Studio» версии от 2015 года.

**Решение:**

```

Using System;
Namespace HelloApp
{
    Class Program
    {
        Static void Main(string[] args)
        {
            Console.WriteLine("Введите число: ");
            Int num1 = Convert.ToInt32(Console.ReadLine());
            Console.WriteLine("Введите число: ");
            Int num2 = Convert.ToInt32(Console.ReadLine());

```

```

If (num1 > num2)
{
Console.WriteLine("Первое число больше второго ");
}
Else If (num1 < num2)
{
Console.WriteLine("Первое число меньше второго ");
}
Else
{
Console.WriteLine("Оба числа равны ");
}
Console.ReadKey();
}
}
}

```

### Задача № 20 «Только нечётные»

**Возраст:** 12-17 лет

**Условие:** из введённого числа, удалить все чётные числа.

**Материально-техническое обеспечение:** ноутбук, программный комплекс «Visual Studio» версии от 2015 года.

**Решение:**

```

using System;
class Program
{
    static void Main(string[] args)
    {
        var n = Console.ReadLine();

        n = n.Replace("0", string.Empty)
            .Replace("2", string.Empty)
            .Replace("4", string.Empty)
            .Replace("6", string.Empty)
            .Replace("8", string.Empty);

        n = n.Length > 0 ? n : "0";
        Console.WriteLine(n);
    }
}

```

*Ведерников Владислав Викторович,  
педагог дополнительного образования  
мобильного ДТ «Кванториум-28»*

### Задача № 21 «Подсчёт суммы»

**Возраст:** 13-17 лет

**Условие:** напишите программу, которая запрашивает у пользователя числа, а затем выводит их сумму.

**Материально-техническое обеспечение:** компьютер или ноутбук с операционной системой, на которой можно установить Python. Клавиатура и мышь для ввода данных и управ-

ления программами. Монитор или экран для отображения кода и результатов выполнения программ.

**Программное обеспечение:**

1. Python: Интерпретатор Python необходим для запуска и выполнения программ на языке Python. Можно скачать с официального сайта Python (<https://www.python.org/>) и установить на компьютер.

2. Текстовый редактор или интегрированная среда разработки (IDE). Для написания кода можно использовать различные редакторы, такие как: Visual Studio Code, PyCharm, Sublime Text, Atom и другие.

3. Консоль или терминал: для запуска программ можно использовать командную строку или терминал, в зависимости от операционной системы.

4. Ноутбук, программный комплекс «Visual Studio» версии от 2015 года.

**Решение:**

```
numbers = input("Введите числа через пробел: ")
total = 0
for num in numbers.split():
    total += float(num)
print("Сумма чисел:", total)
```

## Задача № 22 «Поиск максимума»

**Возраст:** 13-17 лет

**Условие:** напишите программу, которая находит максимальное число из списка.

**Материально-техническое обеспечение:** компьютер или ноутбук с операционной системой, на которой можно установить Python. Клавиатура и мышь для ввода данных и управления программами. Монитор или экран для отображения кода и результатов выполнения программ.

**Программное обеспечение:**

1. Python: Интерпретатор Python необходим для запуска и выполнения программ на языке Python. Можно скачать с официального сайта Python (<https://www.python.org/>) и установить на компьютер.

2. Текстовый редактор или интегрированная среда разработки (IDE). Для написания кода можно использовать различные редакторы, такие как: Visual Studio Code, PyCharm, Sublime Text, Atom и другие.

3. Консоль или терминал: для запуска программ можно использовать командную строку или терминал, в зависимости от операционной системы.

4. Ноутбук, программный комплекс «Visual Studio» версии от 2015 года.

**Решение:**

```
numbers = [3, 7, 2, 9, 1]
max_number = max(numbers)
print("Максимальное число:", max_number)
```

## Задача № 23 «Палиндром»

**Возраст:** 13-17 лет

**Условие:** напишите программу, которая определяет, является ли строка палиндромом (читается одинаково с обеих сторон).

**Материально-техническое обеспечение:** компьютер или ноутбук с операционной системой, на которой можно установить Python. Клавиатура и мышь для ввода данных и управления программами. Монитор или экран для отображения кода и результатов выполнения программ.

**Программное обеспечение:**



1. Python: Интерпретатор Python необходим для запуска и выполнения программ на языке Python. Можно скачать с официального сайта Python (<https://www.python.org/>) и установить на компьютер.

2. Текстовый редактор или интегрированная среда разработки (IDE): для написания кода можно использовать различные редакторы, такие как Visual Studio Code, PyCharm, Sublime Text, Atom и другие.

3. Консоль или терминал: для запуска программ можно использовать командную строку или терминал, в зависимости от операционной системы.

**Решение:**

```
string = input("Введите строку: ")
is_palindrome = all(string[i] == string[-i - 1] for i in range(len(string) // 2))
print("Это палиндром" if is_palindrome else "Это не палиндром")
```

## Задача № 24 «Факториал»

**Возраст:** 13-17 лет

**Условие:** напишите программу для вычисления факториала числа.

**Материально-техническое обеспечение:** компьютер или ноутбук с операционной системой, на которой можно установить Python. Клавиатура и мышь для ввода данных и управления программами. Монитор или экран для отображения кода и результатов выполнения программ.

**Программное обеспечение:**

1. Python: Интерпретатор Python необходим для запуска и выполнения программ на языке Python. Можно скачать с официального сайта Python (<https://www.python.org/>) и установить на компьютер.

2. Текстовый редактор или интегрированная среда разработки (IDE): для написания кода можно использовать различные редакторы, такие как Visual Studio Code, PyCharm, Sublime Text, Atom и другие.

3. Консоль или терминал: для запуска программ можно использовать командную строку или терминал, в зависимости от операционной системы.

4. Ноутбук, программный комплекс «Visual Studio» версии от 2015 года.

**Решение:**

```
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)
num = int(input("Введите число: "))
print("Факториал числа", num, ":", factorial(num))
```

## Задача № 25 «Сортировка списка»

**Возраст:** 13-17 лет

**Условие:** напишите программу для сортировки списка чисел в порядке убывания.

**Материально-техническое обеспечение:** компьютер или ноутбук с операционной системой, на которой можно установить Python. Клавиатура и мышь для ввода данных и управления программами. Монитор или экран для отображения кода и результатов выполнения программ.

**Программное обеспечение:**

1. Python: Интерпретатор Python необходим для запуска и выполнения программ на языке Python. Можно скачать с официального сайта Python (<https://www.python.org/>) и установить на компьютер.

2. Текстовый редактор или интегрированная среда разработки (IDE): для написания кода можно использовать различные редакторы, такие как Visual Studio Code, PyCharm, Sublime Text, Atom и другие.

3. Консоль или терминал: для запуска программ можно использовать командную строку или терминал, в зависимости от операционной системы.

4. Ноутбук, программный комплекс «Visual Studio» версии от 2015 года.

**Решение:**

```
numbers = [3, 1, 4, 1, 5, 9, 2, 6, 5]
sorted_numbers = sorted(numbers, reverse=True)
print("Отсортированный список:", sorted_numbers)
```

## Задача № 26 «Подсчет гласных»

**Возраст:** 13-17 лет

**Условие:** напишите программу, которая запрашивает у пользователя строку и подсчитывает количество гласных букв в этой строке. Гласные буквы в английском алфавите: 'a', 'e', 'i', 'o', 'u', 'y' (в нижнем и верхнем регистрах).

**Материально-техническое обеспечение:** компьютер или ноутбук с операционной системой, на которой можно установить Python. Клавиатура и мышь для ввода данных и управления программами. Монитор или экран для отображения кода и результатов выполнения программ.

**Программное обеспечение:**

1. Python: Интерпретатор Python необходим для запуска и выполнения программ на языке Python. Можно скачать с официального сайта Python (<https://www.python.org/>) и установить на компьютер.

2. Текстовый редактор или интегрированная среда разработки (IDE): для написания кода можно использовать различные редакторы, такие как Visual Studio Code, PyCharm, Sublime Text, Atom и другие.

3. Консоль или терминал: для запуска программ можно использовать командную строку или терминал, в зависимости от операционной системы.

4. Ноутбук, программный комплекс «Visual Studio» версии от 2015 года

**Решение:**

```
def count_vowels(string):
    vowels = "aeiouyAEIOUY"
    count = 0
    for char in string:
        if char in vowels:
            count += 1
    return count
input_string = input("Введите строку: ")
vowel_count = count_vowels(input_string)
print("Количество гласных в строке:", vowel_count)
```

## Программирование на языке C++

*Алексанов Роман Андреевич,  
педагог дополнительного образования  
ДТ «Кванториум-28»*

### Задача № 1 «Работа с вводом/выводом»

**Возраст:** 13-14 лет

**Условие:** напишите программу, которая выводит на экран фразу "Hello, World!"

**Решение:**

```
#include <iostream>
using namespace std;

int main() {
    std::cout << "Hello, World!" << std::endl;
    return 0;
}
```

### Задача № 2 «Работа со стандартными математическими функциями»

**Возраст:** 13-14 лет

**Условие:** напишите программу, которая будет считать и выводить на экран сумму двух чисел.

**Решение:**

```
#include <iostream>
using namespace std;

int main() {
    int a = 10;
    int b = 5;
    int sum = a + b;
    std::cout << "Сумма чисел " << a << " и " << b << " равна " << sum <<
std::endl;
    return 0;
}
```

### Задача № 3 «Работа с операторами сравнения»

**Возраст:** 11-12 лет

**Условие:** напишите программу, которая будет находить максимальное число из трех введенных пользователем.

**Решение:**

```
#include <iostream>
using namespace std;
int main() {
    int a, b, c;
    std::cin >> a >> b >> c;
    int max = (a > b) ? ((a > c) ? a : c) : ((b > c) ? b : c);
    std::cout << "Максимальное число: " << max << std::endl;
    return 0;
}
```

## Задача № 4 «Условные операторы»

**Возраст:** 13-14 лет

**Условие:** напишите программу, которая будет проверять, является ли введенное пользователем число чётным.

**Решение:**

```
#include <iostream>
using namespace std;

int main() {
    int num;
    std::cin >> num;
    if (num % 2 == 0) {
        std::cout << num << " - четное число" << std::endl;
    } else {
        std::cout << num << " - нечетное число" << std::endl;
    }
    return 0;
}
```

## Задача № 5 «Работа с циклом for»

**Возраст:** 13-14 лет

**Условие:** напишите программу, которая будет перемножать все числа от 1 до введенного пользователем.

**Решение:**

```
#include <iostream>
using namespace std;

int main() {
    int n, result = 1;
    std::cin >> n;
    for (int i = 1; i <= n; i++) {
        result *= i;
    }
    std::cout << "Произведение всех чисел от 1 до " << n << " равно " <<
    result << std::endl;
    return 0;
}
```

## Задача № 6 «Работа с пользовательским вводом»

**Возраст:** 13-14 лет

**Условие:** напишите программу, которая будет проверять, является ли введенное пользователем число простым.

**Решение:**

```
#include <iostream>
using namespace std;

int main() {
    int num;
```

```

bool is_prime = true;
std::cin >> num;
for (int i = 2; i <= num / 2; i++) {
    if (num % i == 0) {
        is_prime = false;
        break;
    }
}
if (is_prime) {
    std::cout << num << " - простое число" << std::endl;
} else {
    std::cout << num << " - составное число" << std::endl;
}
return 0;
}

```

### Задача № 7 «Работа с библиотекой математических функций»

**Возраст:** 13-14 лет

**Условие:** напишите программу, которая будет находить корни квадратного уравнения  $ax^2 + bx + c = 0$ .

**Решение:**

```

#include <iostream>
#include <cmath>
using namespace std;

int main() {
    double a, b, c, D, x1, x2;
    std::cin >> a >> b >> c;
    D = b * b - 4 * a * c;
    if (D > 0) {

```

### Задача № 8 «Факториал в C++»

**Возраст:** 13-14 лет

**Условие:** напишите программу, которая будет находить факториал введённого пользователем числа.

**Решение:**

```

#include <iostream>
using namespace std;

int main() {
    int num, fact = 1;
    std::cin >> num;
    for (int i = 1; i <= num; i++) {
        fact *= i;
    }
    std::cout << "Факториал числа " << num << " равен " << fact << std::endl;
    return 0;
}

```

## РАЗДЕЛ 4: «ВИРТУАЛЬНАЯ РЕАЛЬНОСТЬ И ПРОМЫШЛЕННЫЙ ДИЗАЙН»

*Ларин Павел Александрович,  
педагог дополнительного образования  
мобильного ДТ «Кванториум-28»*

### Unity

#### Задача № 1 «Расстановка по пьедесталам»

**Возраст:** 14-17 лет

**Условие:** создать прототип приложения виртуальной реальности, в котором пользователю необходимо будет расставить по «пьедесталам» объекты, которые соответствуют описанию.

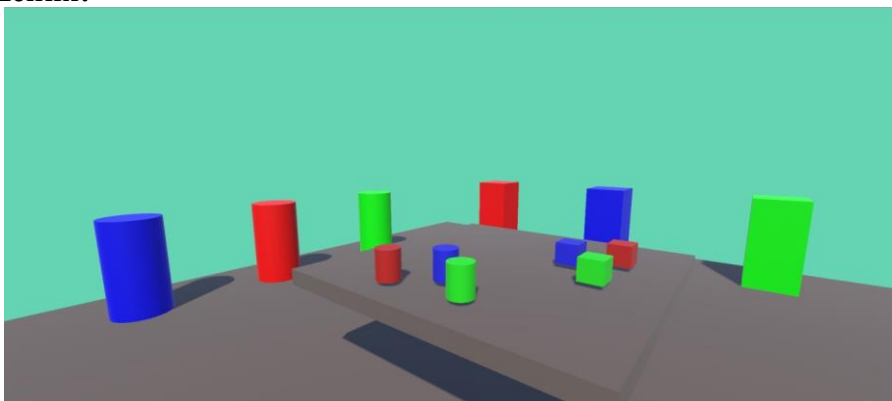
**Материально-техническое обеспечение:**

1. Компьютеры или ноутбуки с ОС Windows 10.
2. Программное обеспечение Unity Engine не старше 2019.4.0.
3. Плагин SteamVR для Unity.
4. Среда разработки Microsoft Visual Studio не старше 2019 года или редактор Microsoft Visual Code.
5. Дистрибутивы Microsoft Visual C++ от 2005 до 2022 года.
6. Дистрибутивы Microsoft DirectX от 9 до 12 версии.
7. OpenVR совместимый шлем виртуальной реальности с контроллерами.
8. Комплект документации для Unity и SteamVR.
9. Доступ в интернет.

**Ход выполнения:**

1. Создать новый проект в Unity Hub.
2. Установить плагин SteamVR при помощи Unity Asset Store или из локального файла, который должен быть заранее скачан на компьютер.
3. Добавить на сцену пол, модель игрока, объекты заданий, подсказки для пользователя и три «пьедестала».
4. При помощи инструментов SteamVR добавить модели игрока, возможность перемещение игрока в пространстве и возможность взаимодействия с объектами.
5. Для объектов заданий добавить физику и обеспечить возможность взять их в «руки» игрока.
6. Создать скрипты для «пьедесталов», который будут определять тип объекта, расположенных на них. В случае правильно расположенного объекта «пьедестал» должен сменить цвет на зелёный, иначе «пьедестал» должен иметь красный цвет.

**Пример решения:**





## Задача № 2 «Мишень»

**Возраст:** 14-17 лет

**Условие:** создать прототип приложения виртуальной реальности, в котором пользователь должен метать предметы в мишень.

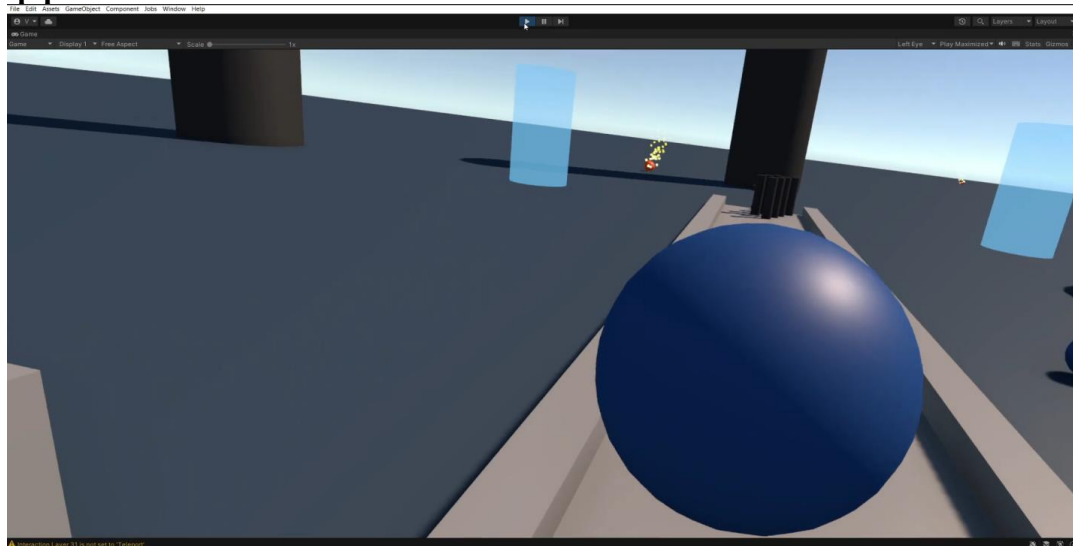
**Материально-техническое обеспечение:**

1. Компьютеры или ноутбуки с ОС Windows 10.
2. Программное обеспечение Unity Engine не старше 2019.4.0.
3. Плагин SteamVR для Unity.
4. Среда разработки Microsoft Visual Studio не старше 2019 года или редактор Microsoft Visual Code.
5. Дистрибутивы Microsoft Visual C++ от 2005 до 2022 года.
6. Дистрибутивы Microsoft DirectX от 9 до 12 версии.
7. OpenVR совместимый шлем виртуальной реальности с контроллерами.
8. Комплект документации для Unity и SteamVR.
9. Доступ в интернет

**Ход выполнения:**

1. Создать проект в Unity Hub.
2. Установить плагин SteamVR при помощи Unity Asset Store или из локального файла, который должен быть заранее скачан на компьютер.
3. Добавить на сцену пол, модель игрока, снаряд для метания, подсказки для пользователя и три мишени.
4. При помощи инструментов SteamVR добавить модели игрока, возможность перемещение игрока в пространстве и возможность взаимодействия с объектами.
5. Для снарядов для метания добавить физику и обеспечить возможность взять их в «руки» игрока.
6. Создать скрипт, который будет определять прохождение снаряда через мишень и менять цвет мишени на зелёный если есть попадание в мишень.

**Пример решения:**



## Задача № 3 «Виртуальный музей»

**Возраст:** 14-17 лет

**Условие:** создать прототип приложения виртуальной реальности, в котором пользователь сможет посетить виртуальный музей 3D-моделей.

**Материально-техническое обеспечение:**

1. Компьютеры или ноутбуки с ОС Windows 10.
2. Программное обеспечение Unity Engine не старше 2019.4.0.
3. Плагин SteamVR для Unity.
4. Среда разработки Microsoft Visual Studio не старше 2019 года или редактор Microsoft Visual Code.
5. Дистрибутивы Microsoft Visual C++ от 2005 до 2022 года.
6. Дистрибутивы Microsoft DirectX от 9 до 12 версии.
7. OpenVR совместимый шлем виртуальной реальности с контроллерами.
8. Комплект документации для Unity и SteamVR.
9. Доступ в интернет

**Ход выполнения:**

1. Создать проект в Unity Hub.
2. Установить плагин SteamVR при помощи Unity Asset Store или из локального файла, который должен быть заранее скачан на компьютер.
3. Добавить на сцену пол, модель игрока и несколько витрин с экспонатами (экспонаты можно скачать из интернета или использовать 3д модели, созданные другими учениками)
4. Добавить скрипт, отображающий название и описание экспоната на витрине при поднесении руки к экспонату.
5. Добавить возможность взять экспонаты в руки и осмотреть их.

**Пример VR музея**



### **Задача № 4 «Приложение для медитации»**

**Условие:** создать прототип приложения виртуальной реальности для медитации.

**Возраст:** 14-17 лет

**Материально-техническое обеспечение:**

1. Компьютеры или ноутбуки с ОС Windows 10.
2. Программное обеспечение Unity Engine не старше 2019.4.0.
3. Плагин SteamVR для Unity.
4. Среда разработки Microsoft Visual Studio не старше 2019 года или редактор Microsoft Visual Code.
5. Дистрибутивы Microsoft Visual C++ от 2005 до 2022 года.
6. Дистрибутивы Microsoft DirectX от 9 до 12 версии.
7. OpenVR совместимый шлем виртуальной реальности с контроллерами.
8. Комплект документации для Unity и SteamVR.
9. Доступ в интернет

**Ход выполнения:**

1. Создать проект в Unity Hub.
2. Установить плагин SteamVR при помощи Unity Asset Store или из локального файла, который должен быть заранее скачан на компьютер.
3. Добавить на сцену пол, модель игрока, найти в Unity Asset Store подходящие ассеты для создания расслабляющей атмосферы.
4. Расставить найденные ассеты по сцене, создав небольшую расслабляющую/умиротворяющую сцену.
5. Найти и добавить на сцену спокойную музыку.

### Пример VR-медитации



## Задача № 5 «Приложение для тренировки реакции»

**Возраст:** 14-17 лет

**Условие:** создать прототип приложения виртуальной реальности для тренировки реакции.

### Материально-техническое обеспечение:

1. Компьютеры или ноутбуки с ОС Windows 10.
2. Программное обеспечение Unity Engine не старше 2019.4.0.
3. Плагин SteamVR для Unity.
4. Среда разработки Microsoft Visual Studio не старше 2019 года или редактор Microsoft Visual Code.
5. Дистрибутивы Microsoft Visual C++ от 2005 до 2022 года.
6. Дистрибутивы Microsoft DirectX от 9 до 12 версии.
7. OpenVR совместимый шлем виртуальной реальности с контроллерами.
8. Комплект документации для Unity и SteamVR.
9. Доступ в интернет

### Ход выполнения:

1. Создать проект в Unity Hub.
2. Установить плагин SteamVR при помощи Unity Asset Store или из локального файла, который должен быть заранее скачан на компьютер.
3. Добавить на сцену пол, модель игрока и модель оружия и мишени.
4. Добавить модели оружия возможность функционал стрельбы.
5. Создать скрипт, который будет активировать мишени для попаданий на короткий период, при успешном попадании мишень станет зелёной, при промахе или раннем выстреле - красной.

## Пример приложения



### Задача № 6 «Приложение для тренировки меткости»

**Возраст:** 14-17 лет

**Условие:** создать прототип приложение виртуальной реальности для тренировки меткости.

**Материально-техническое обеспечение:**

1. Компьютеры или ноутбуки с ОС Windows 10.
2. Программное обеспечение Unity Engine не старше 2019.4.0.
3. Плагин SteamVR для Unity.
4. Среда разработки Microsoft Visual Studio не старше 2019 года или редактор Microsoft Visual Code.
5. Дистрибутивы Microsoft Visual C++ от 2005 до 2022 года.
6. Дистрибутивы Microsoft DirectX от 9 до 12 версии.
7. OpenVR совместимый шлем виртуальной реальности с контроллерами.
8. Комплект документации для Unity и SteamVR.
9. Доступ в интернет

**Ход выполнения:**

1. Создать проект в Unity Hub.
2. Установить плагин SteamVR при помощи Unity Asset Store или из локального файла, который должен быть заранее скачан на компьютер.
3. Добавить на сцену пол, модель игрока и модель оружия и мишени (возможны несколько вариантов мишеней).
4. Добавить модели оружия возможность функционал стрельбы.
5. Создать скрипты, которые будут рассчитывать количество получаемых игроком очков в зависимости от места попадания (чем ближе к центру – тем больше) и считать количество полученных игроком очков.

**Пример решения:**





*Шаповалов Денис Иванович,  
педагог дополнительного образования  
площадки ДТ «Кванториум-28» г. Свободный*

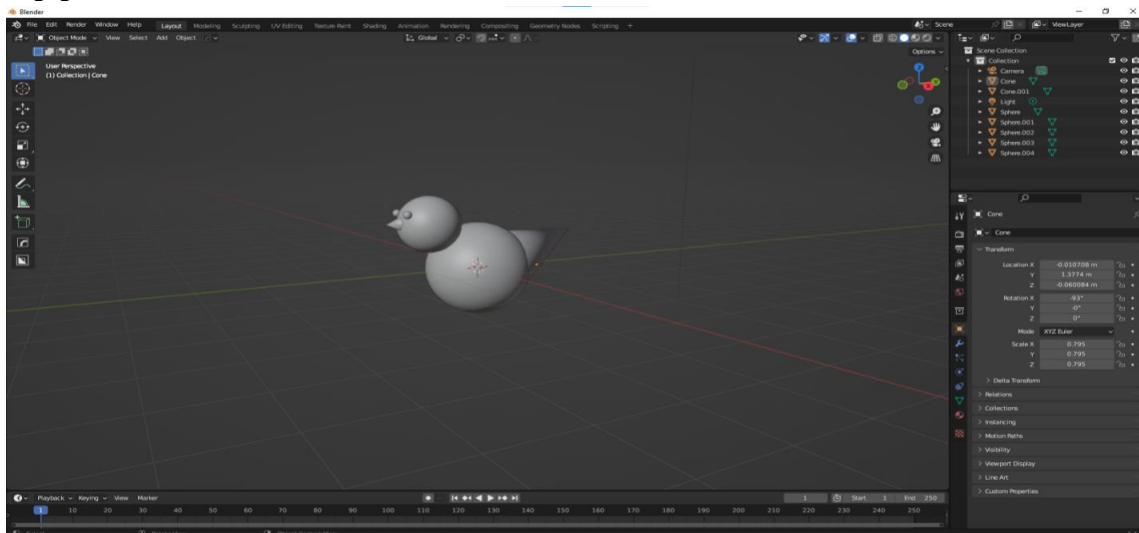
## Blender

### Задача № 1 «Цыплёнок»

**Возраст:** 12-17 лет

**Условие:** показывается проекция сцены, где расположены объекты, из которых создан цыплёнок. Учащиеся должны самостоятельно создать похожую сцену в Blender.

**Пример решения:**



### Задача № 2 «Ромашка»

**Возраст:** 12-17 лет

**Условие:** создайте цветок, используя операции масштабирования поворота и перемещения в Blender.

## Пример решения:



## Задача № 3 «Мебель»

**Возраст:** 12-17 лет

**Условие:** создайте стол и стул, используя инструменты Curve в Blender.

**Ход выполнения:**

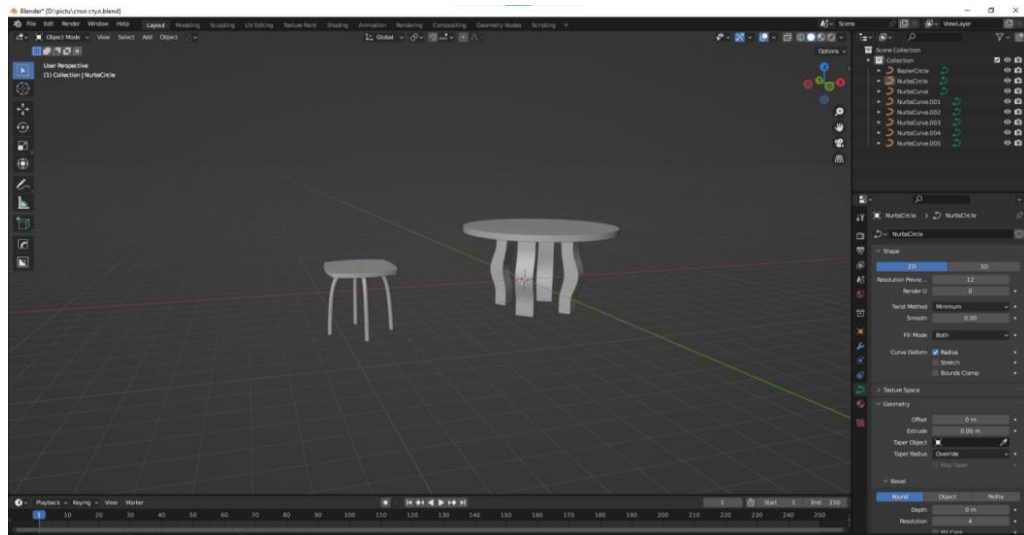
Добавьте на сцену кривую Nurbs:

1. Перейдите в режим edit mode и, используя узлы, очертите контур будущей ножки.
2. В свойствах кривой выберите режим 2D метод заливки Both.
3. В разделе «Геометрия» установите необходимое значение пункта extrude.
4. Сдублируйте, получившуюся ножку в зависимости от количества задуманных ножек.
5. Столешницу можно создать, используя Nurbs Circle и повторив операции 3 и 4;
6. Расставьте ножки и столешницу, чтобы получился стол.

Для создания стула, также используем Nurbs Curve:

1. Добавьте на сцену кривую Nurbs.
2. Перейдите в режим edit mode и, используя узлы, очертите контур будущей ножки.
3. В свойствах кривой, раздел Bevel, измените значение Resolution, чтобы получилась ножка.
4. Сдублируйте получившуюся ножку в зависимости от количества задуманных ножек.
5. Сидение можно создать, используя также Nurbs и повторив операции 2 и 3.
6. Расставьте ножки и столешницу, чтобы получился стол.

## Пример решения:



## Задача № 4 «Тарелка»

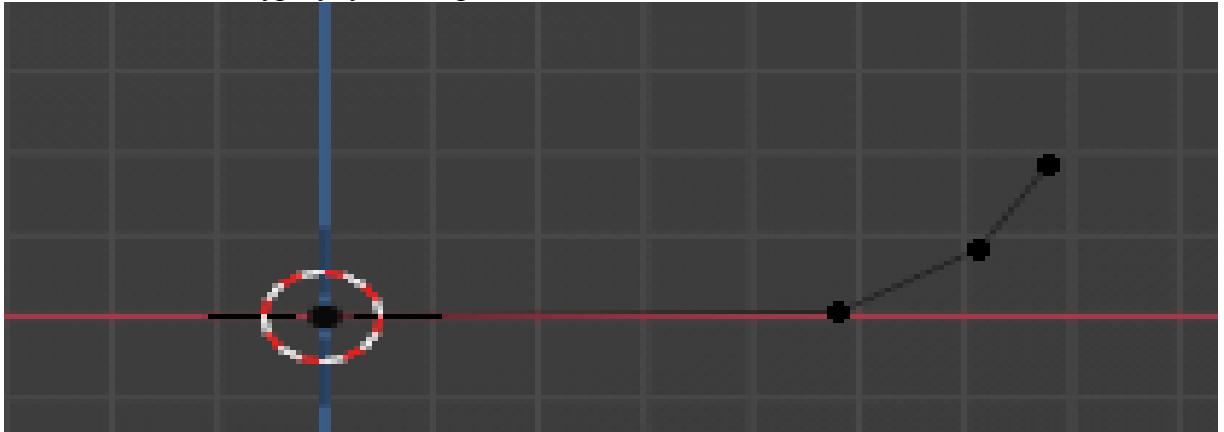
**Возраст:** 12-17 лет

**Условие:** создайте тарелку, используя модификатор screw в Blender.

**Решение:**

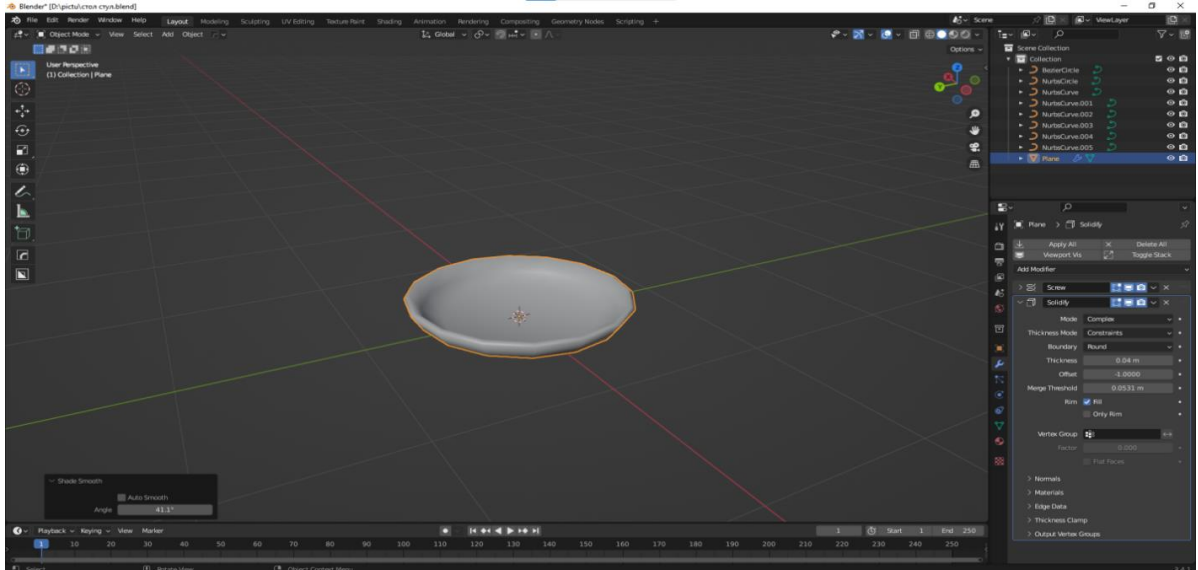
Поместите на сцену плоскость:

1. Перейти в edit mode.
2. Удалить 3 вершины у плоскости и используя оставшуюся вершину и инструмент extrude
3. Создать контур будущей тарелки.



4. Зайти в модификаторы, выбрать модификатор screw по оси Z.
5. Также в модификаторах выбрать solidify, для придания тарелке плотности.

**Пример решения:**



## Задача № 5 «Башня»

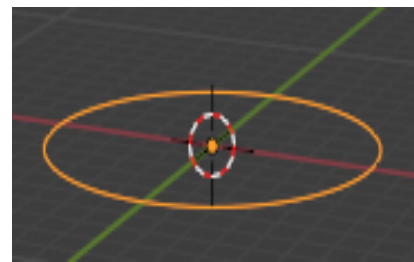
**Возраст:** 12-17 лет

**Условие:** создайте башню, используя инструмент редактирования extrude в Blender.

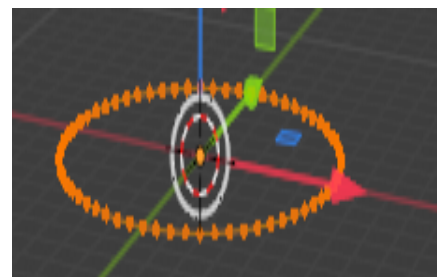
**Ход выполнения:**



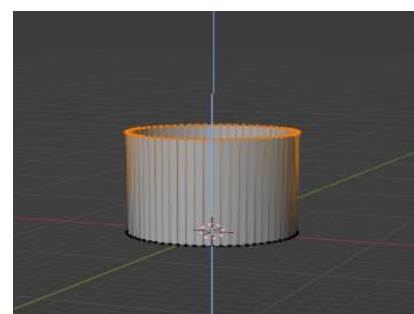
1. Добавляем на сцену окружность.



2. Переходим в режим Edit mode.

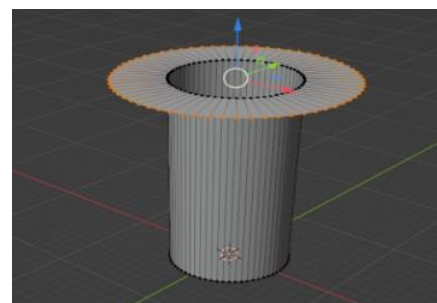


3. Выдавливаем выделенные вершины по оси Z на определенную высоту.



4. Не сбрасывая выделение, снова применяем выдавливание и щелкаем левой кнопкой мышки, не меняя положение вершин.

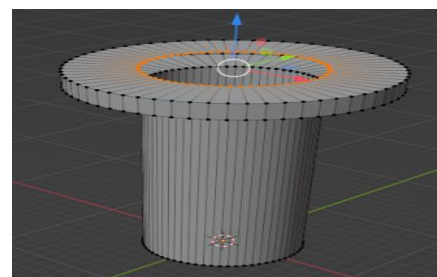
5. Выбираем инструмент масштабирования и масштабируем выделенные вершины наружу цилиндра для создания выступа.



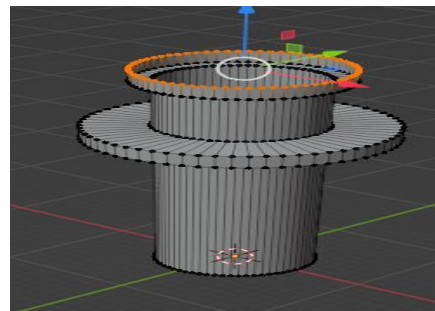
6. Выдавливайте по оси Z, задавая высоту выступа.

7. Не сбрасывая выделение, снова применяем выдавливание и щелкаем левой кнопкой мышки, не меняя положение вершин.

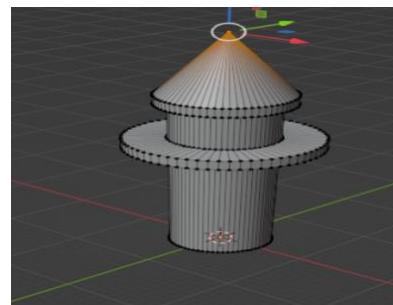
8. Выбираем инструмент масштабирования и масштабируем, выделенные вершины внутрь цилиндра для закрытия выступа.



9. Повторяем операцию из пунктов 3-7.



10. Для создания конусообразной крыши выдавливаем вершины по оси Z и делаем команду Merge (кнопка en M - > All center).



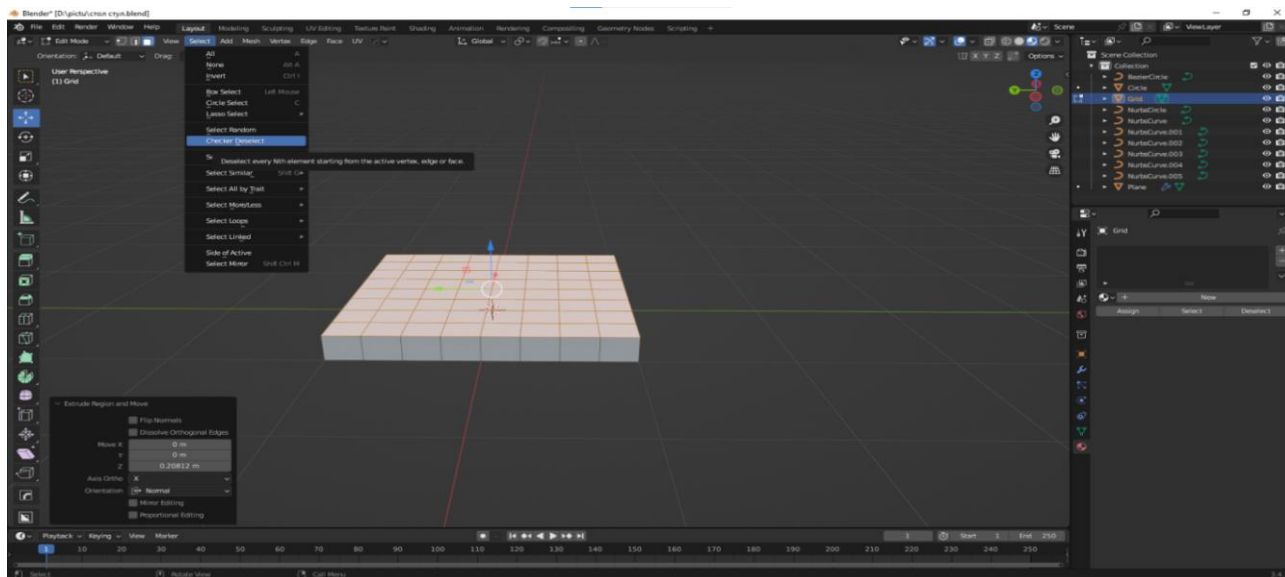
## Задача № 6 «Шахматная доска»

**Возраст:** 12-17 лет

**Условие:** создать шахматную доску, раскрасив полигоны объекта в Blender.

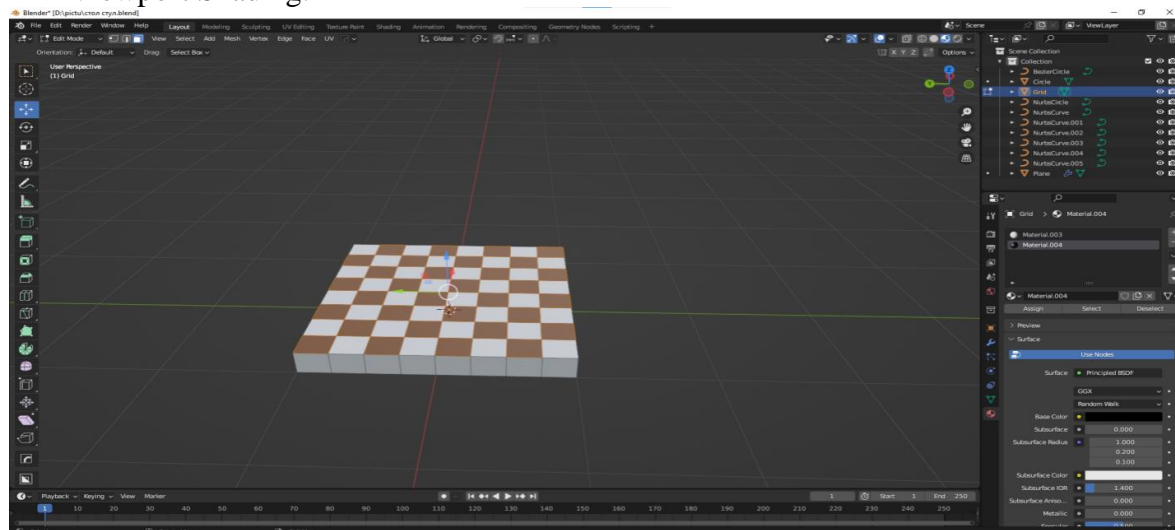
**Ход выполнения:**

1. На сцену добавляем примитив Grid.
2. В специальном меню выставляем количество клеток для шахматной доски (8x8).
3. Переходим в режим edit mode.
4. Выдавливаем по оси Z, задавая толщину доски.
5. В меню режима разделе select выбираем checker deselect.



6. В параметрах объекта свойства материала назначаем основной цвет (белый) нажав кнопку New.
7. В окне выбора материала нажимаем +, чтобы добавить новый 2-ой материал.
8. Нажав кнопку New, выбираем черный цвет и нажимаем кнопку Assign.

9. Выходим из Edit mode и переключаем сцену на просмотр материалов и текстур в Viewport Shading.



Составители: Аргудаева А.А., Дивиченко И.И., Пименов Т.С., Конев П.А., Кулигин А.М., Бондарик Н.В., Лагута Р.Я., Гноевенко И.В., Хмелёв А.П., Ведерников В.В., Алексанов Р.А., Бащенко Б.А., Ларин П.А., Шаповалов Д.И. – Благовещенск, 2024

Общая редакция: Золотарёва Г.В., Рымина Е.Е.

Компьютерный набор и вёрстка: Новиков В.В.

Подписано к печати  
Бумага  
Тираж 25 экз.

Формат  
Усл.печ.л.  
Заказ №

---

675000, г. Благовещенск, ул. Северная, 107  
Государственное автономное учреждение дополнительного профессионального образова-  
ния «Амурский областной институт развития образования»  
Тел.: 8(4162) 226-256, 226-273  
E-mail: amurippk@yandex.ru  
/ Сост.: Золотарёва Г.В., Рымина Е.Е. – Благовещенск, Свободный 2024. – 120 с.