

**Методические рекомендации по решению задач практического
этапа Московского конкурса межпредметных навыков
и знаний «Интеллектуальный мегаполис. Потенциал»
в номинации «Инженерный класс»
по направлению «Авиастроительные классы»**

Пример решения демонстрационного варианта

Кейс №1. (Программирование)

«Программа учёта книг в домашней библиотеке»

Разработать на языке Python программу с консольным интерфейсом для учёта книг в домашней библиотеке. Данные о каждой книге представляют собой совокупность следующих значений:

Номер	0	1	2	3	4
Поле	Название	Автор	Год издания	Язык оригинала	Текстовая заметка

При написании программы каждая команда реализуется в виде отдельной функции, все функции пишутся в одном общем файле вместе с основной программой.

Программа должна позволять выполнять следующие команды:

- **ВВЕСТИ** <Название> <Автор> <Год издания> <Язык оригинала> <Текстовая заметка> – добавляет данные о новой книге с заданными значениями. После выполнения функции в консоль приложения записывается фраза: **ДОБАВЛЕНА НОВАЯ ЗАПИСЬ** <Название>
- **ВЫВЕСТИ** <Название> – выводит в консоль всю информацию о книге с заданным названием. После выполнения функции в консоль приложения записывается фраза: **ИНФОРМАЦИЯ О ЗАПИСИ** <Название> **ПРЕДСТАВЛЕНА ВЫШЕ**
- **СТЕРЕТЬ** <Название> – удаляет книгу с заданным названием. После выполнения функции в консоль приложения записывается фраза: **ЗАПИСЬ** <Название> **УДАЛЕНА**
- **СОРТИРОВАТЬ** <номер_поля_сорт> – выводит в консоль информацию обо всех добавленных книгах, сортируя строки по заданному полю по возрастанию. Если поле строковое, то сортировка должна выполняться по алфавиту. После выполнения функции в консоль приложения записывается фраза: **ДАННЫЕ ОТСОРТИРОВАНЫ ПО** <номер_поля_сорт>.
- **ЗАПИСАТЬ** <имя_файла> – сохраняет добавленные книги в заданный текстовый файл в формате команды **ВВЕСТИ** (с разделением значений пробелами) : <Название> <Автор> <Год издания> <Язык оригинала> <Текстовая заметка>. После выполнения функции в консоль приложения записывается фраза: <имя_файла> **ЗАПИСАН**.
- **СЧИТАТЬ** <имя_файла> – загружает данные о книгах из заданного текстового файла, полностью замещая текущий состав записей о книгах. Считается, что загружаемый файл был создан при вызове команды **ЗАПИСАТЬ**. После выполнения функции в консоль приложения записывается фраза: <имя_файла> **ЗАГРУЖЕН**.

Описание параметров команд:

- <Название> – Название книги БЕЗ разделения пробелами (поле №0), например, ИграПрестолов. Данное значение является уникальным для каждой книги.
- <Автор> – фамилия имя автора книги, строковое значение, не содержащее пробелов (поле №1).
- <Год издания> – год издания книги, натуральное четырёхзначное число (поле №2).

- <Язык оригинала> – язык, на котором первоначально написана книга, строковое значение, не содержащее пробелов (поле №3).
- <Текстовая заметка> – текстовая заметка о книге, строковое значение, не содержащее пробелов (поле №4).
- <номер_поля_сорт> – номер поля, по которому необходимо провести сортировку, целое число с вариантами значения от 0 до 4.
- <имя_файла> – путь (абсолютный или относительный) до текстового файла.

Предполагается, что программе на вход подаются только корректные (не вызывающие ошибок) последовательности команд.

Пример выполнения программы

Ввод	Вывод
ВВЕСТИ СобачьеСердце МихаилБулгаков 1925 Русский АвторКласс!	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ СобачьеСердце
ВВЕСТИ АннаКаренина ЛевТолстой 1877 Русский Неожиданнаяконцовка	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ АннаКаренина ДОБАВЛЕНА НОВАЯ ЗАПИСЬ ТихийДон АннаКаренина ЛевТолстой 1877 Русский Неожиданнаяконцовка
ВВЕСТИ ТихийДон МихаилШолохов 1940 Русский Судьбаказачества	СобачьеСердце МихаилБулгаков 1925 Русский АвторКласс!
СОРТИРОВАТЬ 0	ТихийДон МихаилШолохов 1940 Русский Судьбаказачества
СТЕРЕТЬ ТихийДон	ДАННЫЕ ОТСОРТИРОВАНЫ ПО 0
ВЫВЕСТИ СобачьеСердце	ЗАПИСЬ ТихийДон УДАЛЕНА
ЗАПИСАТЬ db.txt	СобачьеСердце МихаилБулгаков 1925 Русский АвторКласс!
СТЕРЕТЬ АннаКаренина	ИНФОРМАЦИЯ О ЗАПИСИ
СОРТИРОВАТЬ 1	СобачьеСердце ПРЕДСТАВЛЕНА ВЫШЕ db.txt ЗАПИСАН
СЧИТАТЬ db.txt	ЗАПИСЬ АннаКаренина УДАЛЕНА
СОРТИРОВАТЬ 2	СобачьеСердце МихаилБулгаков 1925 Русский АвторКласс!
	ДАННЫЕ ОТСОРТИРОВАНЫ ПО 1 db.txt ЗАГРУЖЕН
	АннаКаренина ЛевТолстой 1877 Русский Неожиданнаяконцовка
	СобачьеСердце МихаилБулгаков 1925 Русский АвторКласс!
	ДАННЫЕ ОТСОРТИРОВАНЫ ПО 2

Критерии оценки:

Реализация каждой из шести команд оценивается отдельно по следующему принципу:

- команда реализована как функция и выполняется корректно – 5 баллов;
- команда реализована не как функция и выполняется корректно – 3 балла;
- команда выполняется некорректно при любом способе реализации – 0 баллов.

№ п/п	Команда	Реализована как функция и выполняется корректно	Реализована не как функция и выполняется корректно	Выполняется не корректно при любом способе реализации
1	ВВЕСТИ	5 баллов	3 балла	0 баллов
2	ВЫВЕСТИ	5 баллов	3 балла	0 баллов
3	СТЕРЕТЬ	5 баллов	3 балла	0 баллов
4	СОРТИРОВАТЬ	5 баллов	3 балла	0 баллов
5	ЗАПИСАТЬ	5 баллов	3 балла	0 баллов
6	СЧИТАТЬ	5 баллов	3 балла	0 баллов
ИТОГО		30 баллов	18 баллов	0 баллов

Решение

Создание 1 функции:

```
def add_record(string: str):
    global data
    string = string.split()
    string[2] = int(string[2])
    data.append(string)
    print(f'ДОБАВЛЕНА НОВАЯ ЗАПИСЬ {string[0]}')
```

Объяснение: Создаём функцию, указываем единственный аргумент, который имеет строчный тип данных. Подключаем глобальную переменную data, для того, чтобы функция могла получить доступ к нашему списку с записями. После входные данные <Название> <Автор> <Год издание> <Язык оригинала> <Текстовая заметка> разбиваем через метод split. После преобразовываем третий столбец к целочисленному типу. После добавляем наши данные в виде списка в список. Переменная data с этого момента будет представлять из себя двумерный список. После пишем строчку вывода, чтобы убедиться в рабочем состоянии функции

Создание 2 функции

```
def print_record(string: str):
    global data
    for rec in data:
        if rec[0] == string:
            print(*rec)
            break
    print(f'ИНФОРМАЦИЯ О ЗАПИСИ {string} ПРЕДСТАВЛЕНА ВЫШЕ')
```

Объяснение: Создаём функцию, указываем единственный аргумент, который имеет строчный тип данных. Подключаем глобальную переменную data, для того, чтобы функция

могла получить доступ к нашему списку с записями. После перебираем каждый список записи. Проверяем, если название записи совпадает с входным значением, то выводим все данные о записи через специальный параметр (*). После пишем строку вывода, чтобы убедиться в рабочем состоянии функции

Создание 3 функции

```
def delete_data(string: str):  
    global data  
    for i in range(len(data)):  
        if data[i][0] == string:  
            data.pop(i)  
            break  
    print(f'ЗАПИСЬ {string} УДАЛЕНА')
```

Объяснение: Создаём функцию, указываем единственный аргумент, который имеет строчный тип данных. Подключаем глобальную переменную `data`, для того, чтобы функция могла получить доступ к нашему списку с записями. После перебираем каждый список записи через доступ по индексу. Проверяем, если название записи совпадает с входным значением, то удаляем запись под этим номером. После пишем строку вывода, чтобы убедиться в рабочем состоянии функции

Создание 4 функции

```
def sort_data(string: str):  
    global data  
    string = int(string)  
    data.sort(key = lambda data: data[string])  
    for rec in data:  
        rec = list(map(str, rec))  
        print(' '.join(rec))  
    print(f'ДААННЫЕ ОТСОРТИРОВАНЫ ПО {string}')
```

Создание 5 функции

```
def save_file(string: str):  
    global data  
    f = open(string, 'w')  
    for i in range(len(data)):  
        rec = list(map(str, data[i]))
```

```
f.write(' '.join(rec)+'\n')
f.close()
print(f'{string} ЗАПИСАН')
```

Создание 6 функции

```
def load_file(string: str):
    global data
    data = []
    f = open(string, 'r')
    for line in f.readlines():
        rec = line.split()
        rec[2] = int(rec[2])
        data.append(rec)
    f.close()
    print(f'{string} ЗАГРУЖЕН')
```

Общий вид программы:

```
data = [ ] #Создание общего списка хранения записей
def add_record(string: str):
    global data
    string = string.split()
    string[2] = int(string[2])
    data.append(string)
    print(f'ДОБАВЛЕНА НОВАЯ ЗАПИСЬ {string[0]}')

def print_record(string: str):
    global data
    for rec in data:
        if rec[0] == string:
            print(*rec)
    print(f'ИНФОРМАЦИЯ О ЗАПИСИ {string} ПРЕДСТАВЛЕНА ВЫШЕ')
```

```

def delete_data(string: str):
    global data
    for i in range(len(data)):
        if data[i][0] == string:
            data.pop(i)
            break
    print(f'ЗАПИСЬ {string} УДАЛЕНА')

def sort_data(string: str):
    global data
    string = int(string)
    data.sort(key = lambda data: data[string])
    for rec in data:
        rec = list(map(str, rec))
        print(' '.join(rec))
    print(f'ДААННЫЕ ОТСОРТИРОВАНЫ ПО {string}')

def save_file(string: str):
    global data
    f = open(string, 'w')
    for i in range(len(data)):
        rec = list(map(str, data[i]))
        f.write(' '.join(rec)+'\n')
    f.close()
    print(f'{string} ЗАПИСАН')

def load_file(string: str):
    global data
    data = [ ]
    f = open(string, 'r')
    for line in f.readlines():
        rec = line.split()

```

```
rec[2] = int(rec[2])
data.append(rec)
f.close()
print(f {string} ЗАГРУЖЕН')
```

```
func = {"ВВЕСТИ": add_record,
        "ВЫВЕСТИ": print_record,
        "СТЕРЕТЬ": delete_data,
        "СОРТИРОВАТЬ": sort_data,
        "ЗАПИСАТЬ": save_file,
        "СЧИТАТЬ": load_file}
```

```
while True:
    string = input()
    command = string.split()[0]
    info = ' '.join(string.split()[1:])
    func[command](info)
```

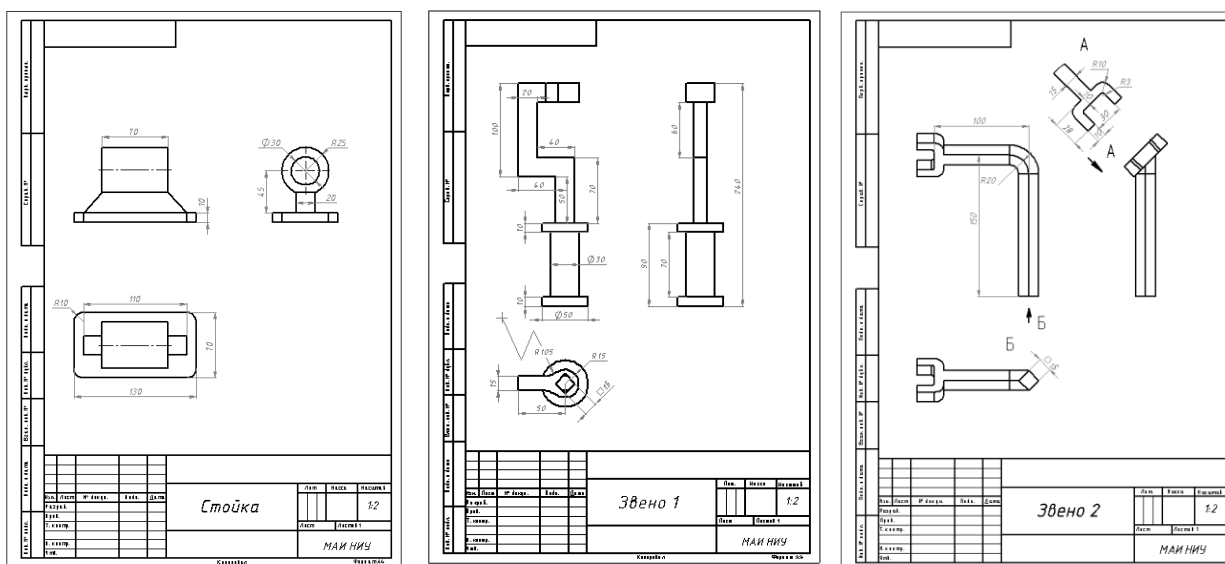

Кейс №2. (3D-моделирование и 3D-печать)

Кейс 2 содержит 3 задания, согласно которым, экзаменуемым предлагается создать прототип робота-манипулятора по его кинематической схеме. Для успешного выполнения задания экзаменуемым необходимо применить навыки конструирования и изготовления трехмерных объектов с помощью 3D принтера, работающего по технологии FDM.

Обратимся к материалам демонстрационного варианта.

Задание 1

Постройте 3D модели звеньев робота согласно чертежам, представленным на рисунках 1 – 3. Экспортируйте детали в формат .stl. – 15 баллов.



Рисунки 1 – 3

Задание 2

Соберите детали из 1ого задания в сборку согласно кинематической схеме, приведенной на рисунке 4. Пересечения деталей не допускается. Экспортируйте сборку в формат .stl. – 10 баллов.

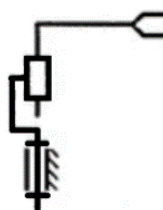


Рисунок 4

Задание 3

Создайте сборку, имитирующую рациональное расположение и ориентацию деталей при печати на 3D принтере на его рабочем столе. За поверхность рабочего стола принтера принимается базовая горизонтальная плоскость (Сверху. Тор, Вид сверху – в зависимости от выбранной САПР) Критерием рациональности выступает минимизация количества поддержек. Экспортируйте сборку в формат .stl. – 5 баллов.

Критерии снижения оценки выполненных заданий

Критерий	Количество снижаемых баллов
Построенная деталь не соответствует чертежу	5 (за каждую деталь)
Построенные детали не сопрягаются без пересечений (интерференция, не совпадают сопрягаемые размеры)	5
Детали собраны в сборку неверно (детали сопряжены неверно)	5
Расположение и ориентация деталей выбраны таким образом, что количество поддержек не минимально	5

РЕКОМЕНДАЦИИ

Для успешного выполнения задания экзаменуемым желательно ознакомиться со следующими темами инженерных дисциплин:

- **САПР:** системы автоматизированного проектирования, 2D эскиз (набросок, sketch), бобышки (тела, body): вытянутые, повернутые, «по траектории»

Для построения деталей с помощью САПР следует придерживаться следующей последовательности действий:

1. Общая оценка всей детали с целью мысленного разбиения ее на отдельные тела;
2. Построение 2D эскиза для одного тела;
3. Построение тела на основе данного 2D эскиза;
4. Повторение шагов 2 и 3 для каждого тела, определенного на 1ом шаге.

Результатом выполнения заданий являются файлы в формате .stl, количество которых для каждого задания:

- Задание 1 - 3 файла;
- Задание 2 - 1 файл;
- Задание 3 - 1 файл.

Таким образом, если экзаменуемый выполнил все 3 задания, то на проверку он должен отправить всего 5 файлов.

Для конвертации готовых деталей в формат .stl, в зависимости от выбранной САПР, возможны 2 подхода:

- Нажатие «Сохранить как» и выбор указанного формата в окне Проводника файловой системы компьютера;
- Экспорт и выбор указанного формата в интерфейсе САПР.

Особое внимание требуется уделить построению именно 2D эскиза. Как правило, именно его размеры и геометрические взаимосвязи служат надежным инструментом при однозначном определении формы и размеров деталей. На рисунке 5 приведен пример определенного эскиза, использованного при построении одного из звеньев демонстрационного варианта.

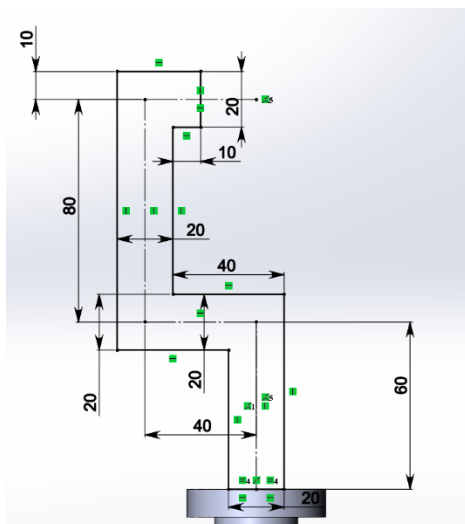


Рисунок 5

Здесь использована в том числе вспомогательная геометрия, которая в некоторых случаях может упростить процесс построения.

При построении эскиза рекомендуется сначала определить его геометрические взаимосвязи, уточнив его форму, а затем образмерить его с помощью размеров, указанных на чертежах.

При выполнении данного задания, следует учитывать, что чертежи содержат полную информацию о форме и размерах звеньев робота, в том числе стойки. При построении моделей деталей возможна любая ориентация их в пространстве.

- **Кинематические пары:** условные обозначения, доступные степени свободы, возможные реализации «в железе».

Абсолютно свободное тело в пространстве имеет 6 степеней свободы: 3 поступательных – то есть допускается перемещение вдоль взаимно перпендикулярных осей X, Y, Z и 3 вращательных – то есть допускается вращение вокруг этих же осей.

На рисунке 6 приведены схемы кинематические пар с указанием их степеней свободы. Это значит, что каждая схема содержит 2 звена и суть схемы заключается в определении, какие ограничения (связи) накладывает сопряжение этих звеньев друг на друга.

Схематическое изображение кинематических пар	Условное изображение по ГОСТ ЕСКД	Число связей S	Степень свободы W
<p>Шар на плоскости</p> 		1	5
<p>Цилиндр на плоскости</p> 		2	4
<p>Призма на плоскости</p> 		3	3
<p>Сферический шарнир</p> 		3	3
<p>Цилиндрическая пара</p> 		4	2
<p>Вращательная пара</p> 		5	1
<p>Поступательная пара</p> 		5	1
<p>Винтовая пара</p> 		5	1

Рисунок 6

Для понимания схем следует представить, что одно из звеньев жестко закреплено и не может ни перемещаться, ни вращаться. Это значит, что у нее есть 0 степеней свободы. Второе звено сопряжено с первым заданную кинематическую пару. И в таком случае у второго звена остается столько степеней свободы, сколько указано на схеме.

- **3D печать по технологии FDM:** основные принципы перемещения рабочего сопла принтера – при печати одного слоя, при переходе от одного слоя к другому, использование поддержек.

Указанная технология 3D печати подразумевает, что рабочее сопло принтера при печати каждого слоя обходит сначала по контуру слоя, а потом заполняет внутреннюю область этого контура сеткой. Это хорошо видно на недопечатанных деталях, как показано на рисунке 7.

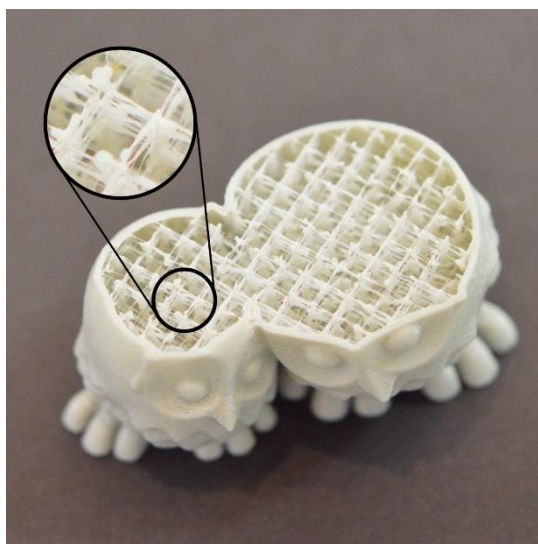


Рисунок 7

Таким образом, новый слой начинает печататься с краев, и каждому новому слою необходимо иметь меньшую площадь, чем предыдущий, чтобы «не свисать» с него. Однако, это не всегда так. Детали, расширяющиеся вверх, могут быть испорчены при печати ввиду вытекания филамента, если расширение краев происходит слишком резко. Для того, чтобы избежать этого, используются поддержки при печати, задаваемые в настройках программы, подготавливающей деталь к печати – программе-слайсеру. Поэтому, добиться качественной печати деталей сложных форм возможно, как используя широкий массив поддержек, либо располагая детали на рабочем столе принтера так, чтобы как можно большее количество последующих слоев печати имело площадь меньшую, чем предыдущие слои.

РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ

На рисунках 8 и 9 приведены примеры успешного выполнения заданий кейса №2.

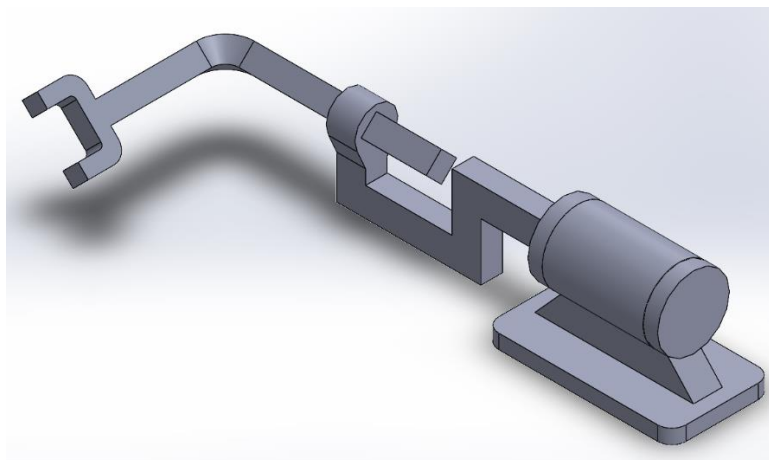


Рисунок 8

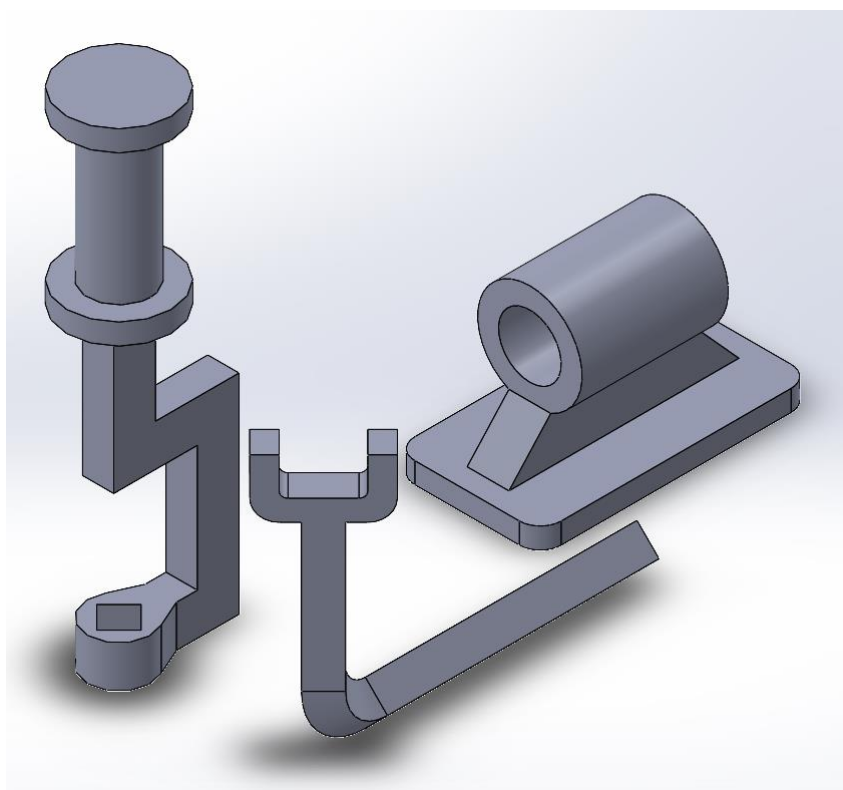


Рисунок 9