

**Номинация «Инженерный класс»**  
**Направление «Авиастроительные классы»**  
**Этап Практический**

**Вариант 1**

**Кейс №1.**

«Программа учёта химических реагентов на складах»

Разработать на языке Python программу с консольным интерфейсом для учёта химических реагентов в лаборатории. Данные о каждом реагенте представляют собой совокупность следующих значений:

<b>Номер</b>	0	1	2	3	4
<b>Поле</b>	Название	Молярная масса	Запас в граммах	Вид	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:

1. **ВПИСАТЬ** <Название> <Молярная масса> <Запас в граммах> <Вид> <Текстовая заметка> – добавляет данные о новом реагенте с заданными значениями. После выполнения функции в консоль приложения записывается фраза: **ДОБАВЛЕНА НОВАЯ ЗАПИСЬ** <Название>
2. **ОТОБРАЗИТЬ** <Название> – выводит в консоль всю информацию о реагентах с заданным названием. После выполнения функции в консоль приложения записывается фраза: **ИНФОРМАЦИЯ О ЗАПИСИ** <Название> **ПРЕДСТАВЛЕНА ВЫШЕ**
3. **УБРАТЬ** <Название> – удаляет реагент с заданным названием. После выполнения функции в консоль приложения записывается фраза: **ЗАПИСЬ** <Название> **УДАЛЕНА**
4. **УПОРЯДОЧИТЬ** <номер\_поля\_сорт> – выводит в консоль информацию обо всех добавленных реагентах, 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>.
5. **СОЗДАТЬ\_ФАЙЛ** <имя\_файла> – сохраняет добавленные реагенты в заданный текстовый файл в формате команды **ВПИСАТЬ** (с разделением значений пробелами): <Название> <Молярная масса> <Запас в граммах> <Вид> <Текстовая заметка>. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> **ЗАПИСАН**.
6. **ПРОЧИТАТЬ\_ФАЙЛ** <имя\_файла> – загружает данные о реагентах из заданного текстового файла, полностью замещая текущий состав записей о реагентах. Считается, что загружаемый файл был создан при вызове команды **СОЗДАТЬ\_ФАЙЛ**. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> **ЗАГРУЖЕН**.

Описание параметров команд:

- <Название> – Название реагента, БЕЗ разделения пробелами (поле №0), например, ХлоридНатрия. Данное значение является уникальным для каждого реагента.
- <Молярная масса> – характеристика реагента, вещественное значение, (поле №1).
- <Запас в граммах> – количество реагента, натуральное число (поле №2).
- <Вид> – классификация реагента, строковое значение, не содержащее пробелов (поле №3).
- <Текстовая заметка> – текстовая заметка о реагенте, 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
ВПИСАТЬ БромидКалия 119.002 115 Неорганическая БелыйПорошок	ДОБАВЛЕНА НОВЫЙ ЗАПИСЬ БромидКалия
ВПИСАТЬ ХлоридЖелеза3 58.44 25 Неорганическая БелыйКристалл	ДОБАВЛЕНА НОВЫЙ ЗАПИСЬ ХлоридЖелеза3
ВПИСАТЬ Глюкоза 180.156 200 Органическая Сладкая	ДОБАВЛЕНА НОВЫЙ ЗАПИСЬ Глюкоза ХлоридЖелеза3 58.44 25 Неорганическая БелыйКристалл
УПОРЯДОЧИТЬ 2	БромидКалия 119.002 115
УБРАТЬ Глюкоза	Неорганическая БелыйПорошок
ОТОБРАЗИТЬ БромидКалия	Глюкоза 180.156 200 Органическая
СОЗДАТЬ_ФАЙЛ reactives.txt	Сладкая
УБРАТЬ ХлоридЖелеза3	ДААННЫЕ ОТСОРТИРОВАНЫ ПО 2
УПОРЯДОЧИТЬ 1	ЗАПИСЬ Глюкоза УДАЛЕНА
ПРОЧИТАТЬ_ФАЙЛ reactives.txt	БромидКалия 119.002 115
УПОРЯДОЧИТЬ 0	Неорганическая БелыйПорошок ИНФОРМАЦИЯ О ЗАПИСИ БромидКалия ПРЕДСТАВЛЕНА ВЫШЕ reactives.txt ЗАПИСАН ЗАПИСЬ ХлоридЖелеза3 УДАЛЕН БромидКалия 119.002 115 Неорганическая БелыйПорошок ДААННЫЕ ОТСОРТИРОВАНЫ ПО 1 reactives.txt ЗАГРУЖЕН БромидКалия 119.002 115 Неорганическая БелыйПорошок ХлоридЖелеза3 58.44 25 Неорганическая БелыйКристалл ДААННЫЕ ОТСОРТИРОВАНЫ ПО 0

### 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	ВПИСАТЬ	5 баллов	3 балла	0 баллов
2	ОТОБРАЗИТЬ	5 баллов	3 балла	0 баллов
3	УБРАТЬ	5 баллов	3 балла	0 баллов
4	УПОРЯДОЧИТЬ	5 баллов	3 балла	0 баллов
5	СОЗДАТЬ ФАЙЛ	5 баллов	3 балла	0 баллов
6	ПРОЧИТАТЬ ФАЙЛ	5 баллов	3 балла	0 баллов
<b>ИТОГО</b>		<b>30 баллов</b>	<b>18 баллов</b>	<b>0 баллов</b>

### Кейс №2.

#### *Задание 1*

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

#### *Задание 2*

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

#### *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. Экпортируйте сборку в формат .stl. – 5 баллов.

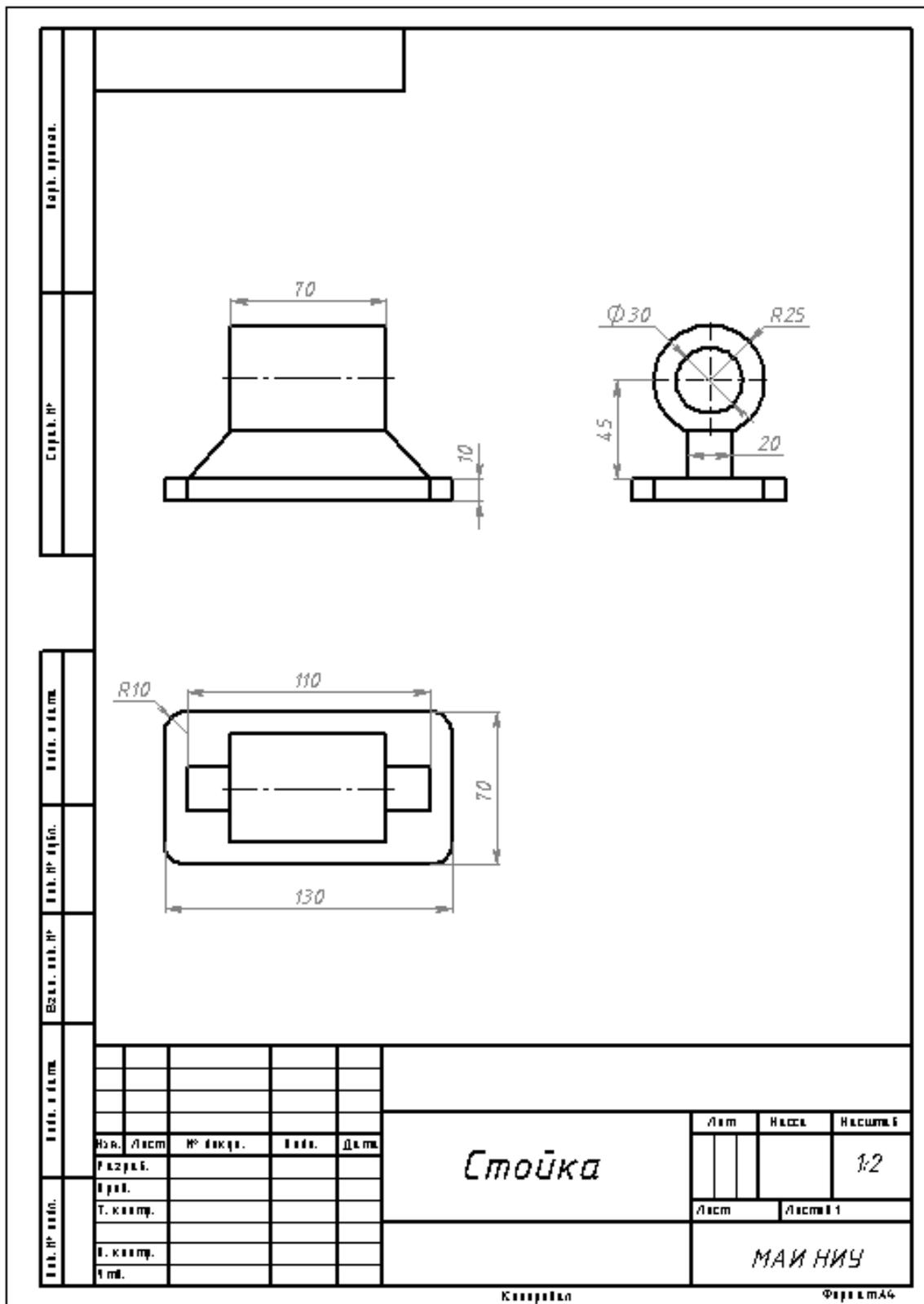


Рисунок 1

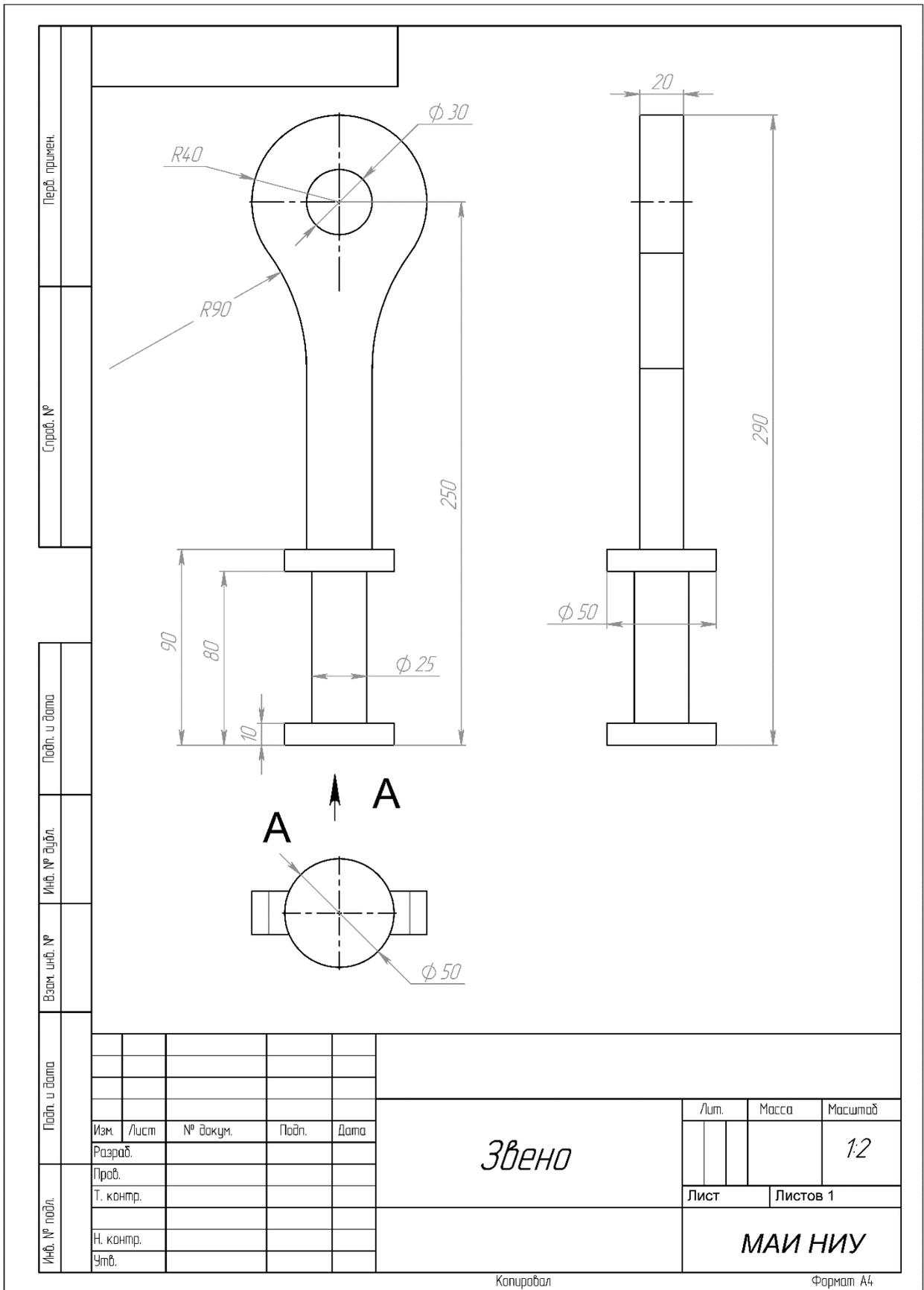


Рисунок 2



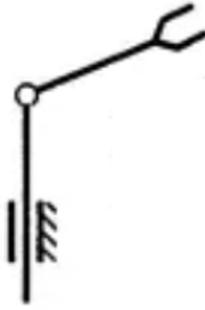


Рисунок 4

## Вариант 2

### Кейс №1.

#### «Программа учёта исторических событий стран»

Разработать на языке Python программу с консольным интерфейсом для хранения записей о ключевых исторических событиях. Данные о каждом событии представляют собой совокупность следующих значений:

Номер	0	1	2	3	4
Поле	Название	Страна	Часть света	Дата	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:

1. **ЗАДАТЬ** <Название> <Страна> <Часть света> <Дата> <Текстовая заметка> – добавляет данные о новом историческом событии с заданными значениями. После выполнения функции в консоль приложения записывается фраза: **ДОБАВЛЕНА НОВАЯ ЗАПИСЬ** <Название>
2. **НАЙТИ** <Название> – выводит в консоль всю информацию о событии с заданным названием. После выполнения функции в консоль приложения записывается фраза: **ИНФОРМАЦИЯ О ЗАПИСИ** <Название> **ПРЕДСТАВЛЕНА ВЫШЕ**
3. **ИСКЛЮЧИТЬ** <Название> – удаляет событие с заданным названием. После выполнения функции в консоль приложения записывается фраза: **ЗАПИСЬ** <Название> **УДАЛЕНА**
4. **СПИСОК** <номер\_поля\_сорт> – выводит в консоль информацию обо всех добавленных событиях, 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>.

5. ЭКСПОРТ <имя\_файла> – сохраняет добавленные события в заданный текстовый файл в формате команды ЗАДАТЬ (с разделением значений пробелами): <Название> <Страна> <Часть света> <Дата> <Текстовая заметка>. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> ЗАПИСАН
6. ИМПОРТ <имя\_файла> – загружает данные о событиях из заданного текстового файла, полностью замещая текущий состав записей об исторических событиях. Считается, что загружаемый файл был создан при вызове команды ЭКСПОРТ. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> ЗАГРУЖЕН.

Описание параметров команд:

- <Название> – название события, БЕЗ разделения пробелами (поле №0), например, КрещениеРуси. Данное значение является уникальным для каждого события.
- <Страна> – страна, к которой относится событие, строковое значение (поле №1).
- <Часть света> – место на планете, где произошло событие, строковое значение, не содержащее пробелов (поле №2).
- <Дата> – год, когда произошло событие, натуральное четырёхзначное число (поле №3).
- <Текстовая заметка> – текстовая заметка о событии, 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
ЗАДАТЬ ПокорениеСибири Россия Азия 1582 Быстроепокорение	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ ПокорениеСибири
ЗАДАТЬ ПадениеПольши Польша Европа 1795 Внезапно	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ ПадениеПольши
ЗАДАТЬ ОсвобождениеКрестьян Россия Евразия 1861 Свободныелюди	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ ОсвобождениеКрестьян
СПИСОК 3	ПокорениеСибири Россия Азия 1582 Быстроепокорение
ИСКЛЮЧИТЬ ПадениеПольши	ПадениеПольши Польша Европа 1795 Внезапно
НАЙТИ ПокорениеСибири	ОсвобождениеКрестьян Россия Евразия 1861
ЭКСПОРТ history.txt	Свободныелюди
ИСКЛЮЧИТЬ ОсвобождениеКрестьян	ДАННЫЕ ОТСОРТИРОВАНЫ ПО 3
СПИСОК 1	ЗАПИСЬ ПадениеПольши УДАЛЕНА
ИМПОРТ history.txt	ПокорениеСибири Россия Азия 1582
СПИСОК 0	Быстроепокорение ИНФОРМАЦИЯ О ЗАПИСИ ПокорениеСибири ПРЕДСТАВЛЕНА ВЫШЕ

	history.txt ЗАПИСАН ЗАПИСЬ ОсвобождениеКрестьян УДАЛЕНА ПокорениеСибири Россия Азия 1582 Быстроепокорение ДАННЫЕ ОТСОРТИРОВАНЫ ПО 1 history.txt ЗАГРУЖЕН ОсвобождениеКрестьян Россия Евразия 1861 Свободныелюди ПокорениеСибири Россия Азия 1582 Быстроепокорение ДАННЫЕ ОТСОРТИРОВАНЫ ПО 0
--	---

### 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 не корректно 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	ЗАДАТЬ	5 баллов	3 балла	0 баллов
2	НАЙТИ	5 баллов	3 балла	0 баллов
3	ИСКЛЮЧИТЬ	5 баллов	3 балла	0 баллов
4	СПИСОК	5 баллов	3 балла	0 баллов
5	ЭКСПОРТ	5 баллов	3 балла	0 баллов
6	ИМПОРТ	5 баллов	3 балла	0 баллов
<b>ИТОГО</b>		<b>30 баллов</b>	<b>18 баллов</b>	<b>0 баллов</b>

## Кейс №2.

### Задание 1

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

### 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.

### 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.

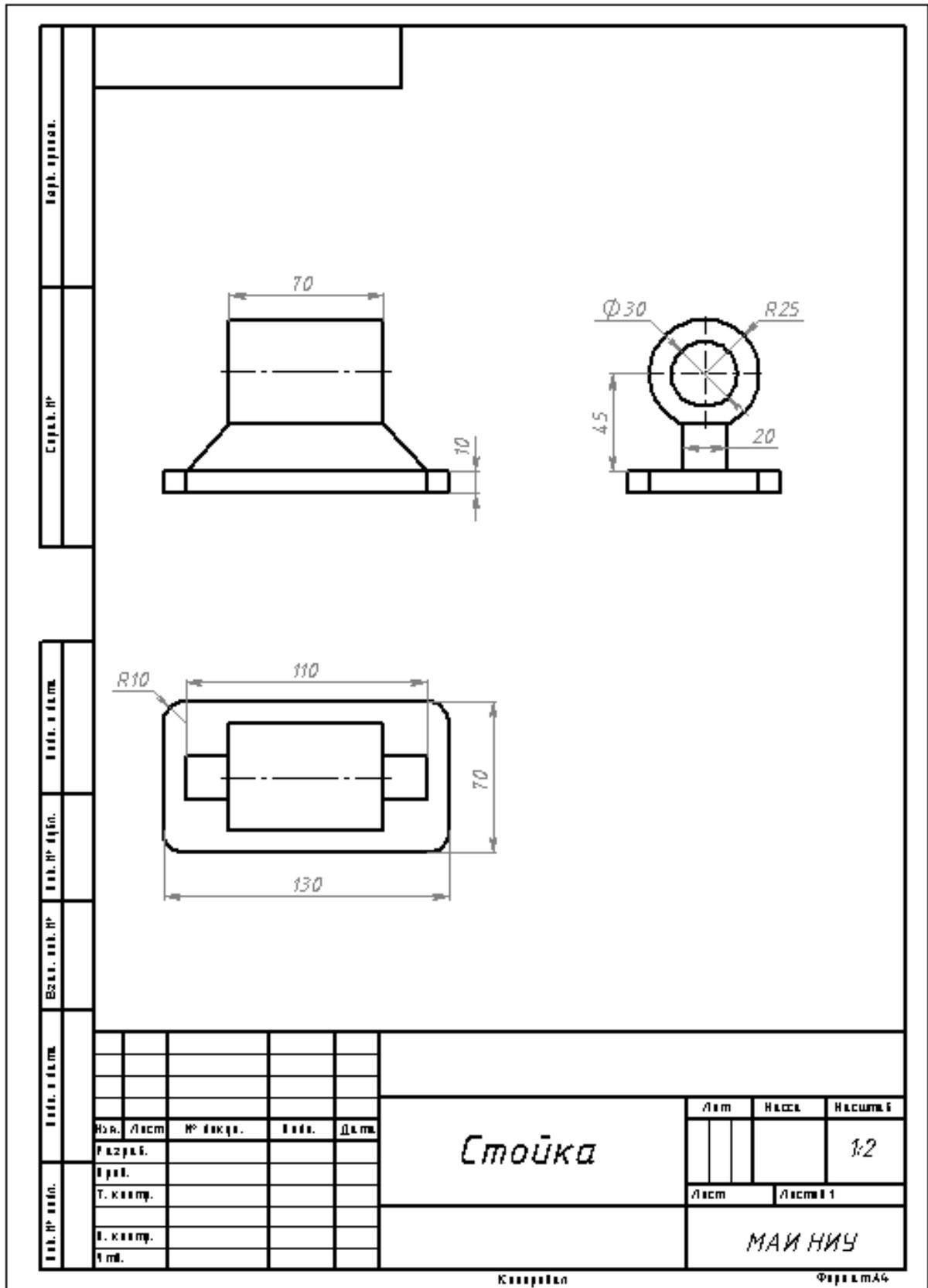


Рисунок 1

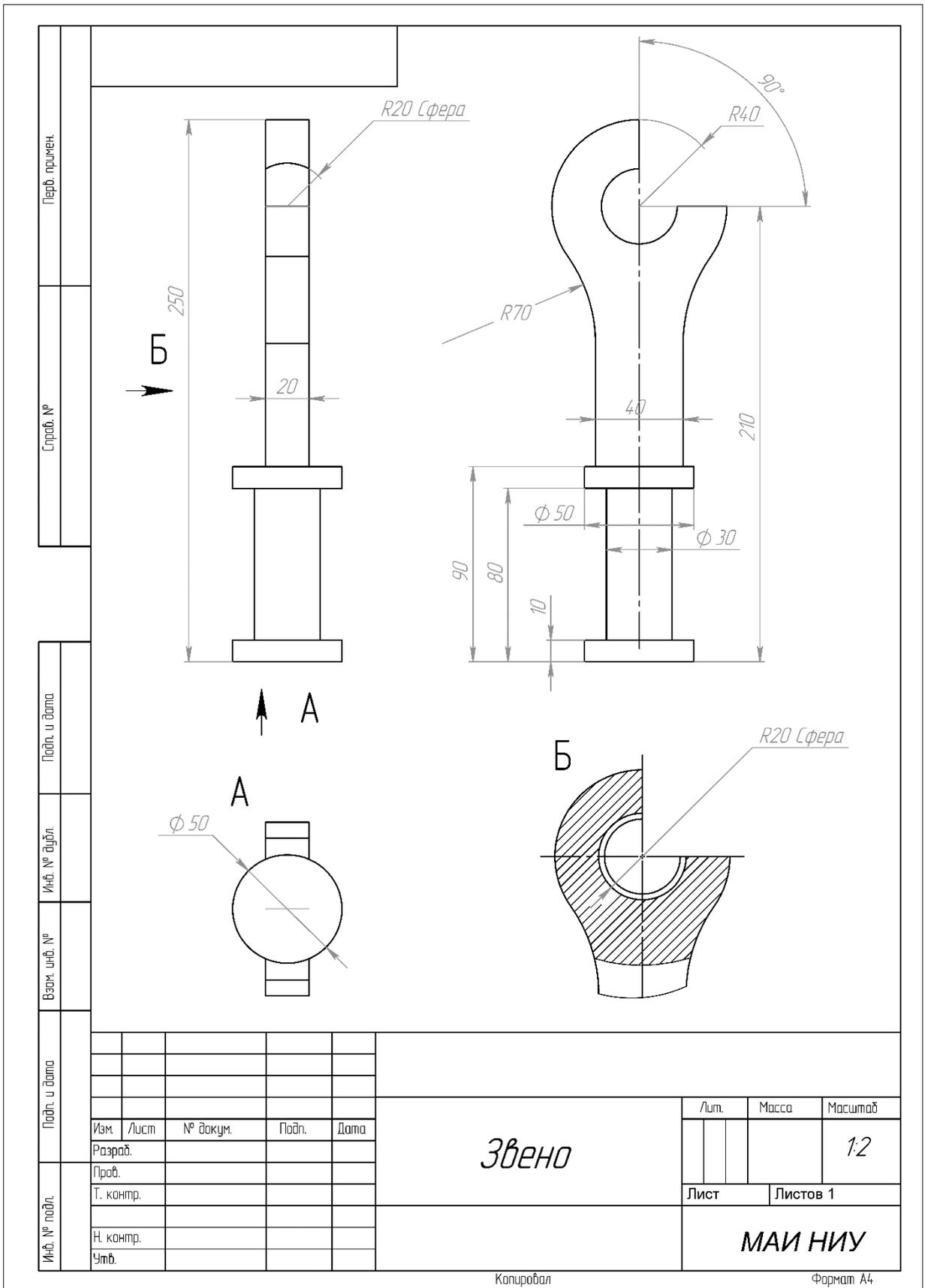


Рисунок 2





Рисунок 4

### Вариант 3

#### Кейс №1.

«Программа учёта ингредиентов в ресторане»

Разработать на языке Python программу с консольным интерфейсом для учёта ингредиентов. Данные о каждом ингредиенте представляют собой совокупность следующих значений:

Номер	0	1	2	3	4
Поле	Название	Цвет	Вкус	Запас в килограммах	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:

1. **ВВЕСТИ** <Название> <Цвет> <Вкус> <Запас в килограммах> <Текстовая заметка> – добавляет данные о новом ингредиенте с заданными значениями. После выполнения функции в консоль приложения записывается фраза: **ДОБАВЛЕНА НОВАЯ ЗАПИСЬ** <Название>
2. **ОТОБРАЗИТЬ** <Название> – выводит в консоль всю информацию об ингредиенте с заданным названием. После выполнения функции в консоль приложения записывается фраза: **ИНФОРМАЦИЯ О ЗАПИСИ** <Название> **ПРЕДСТАВЛЕНА ВЫШЕ**
3. **ИСКЛЮЧИТЬ** <Название> – удаляет ингредиент с заданным названием. После выполнения функции в консоль приложения записывается фраза: **ЗАПИСЬ** <Название> **УДАЛЕНА**
4. **СОРТИРОВАТЬ** <номер\_поля\_сорт> – выводит в консоль информацию обо всех добавленных ингредиентах, 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>.

5. СОЗДАТЬ\_ФАЙЛ <имя\_файла> – сохраняет добавленные ингредиенты в заданный текстовый файл в формате команды ВВЕСТИ (с разделением значений пробелами): <Название> <Цвет> <Вкус> <Запас в килограммах> <Текстовая заметка>. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> ЗАПИСАН.
6. ИМПОРТ <имя\_файла> – загружает данные об ингредиентах из заданного текстового файла, полностью замещая текущий состав записей об ингредиентах. Считается, что загружаемый файл был создан при вызове команды СОЗДАТЬ\_ФАЙЛ. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> ЗАГРУЖЕН.

Описание параметров команд:

- <Название> – Название ингредиента, БЕЗ разделения пробелами (поле №0), например, ЧернаяСмородина. Данное значение является уникальным для каждого ингредиента.
- <Цвет> – характеристика ингредиента, строковое значение, не содержащее пробелов (поле №1).
- <Вкус> – характеристика ингредиента, строковое значение, не содержащее пробелов (поле №2).
- <Запас в килограммах> – количество ингредиента, натуральное число (поле №3).
- <Текстовая заметка> – текстовая заметка об ингредиенте, 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 Морковка оранжевый сладкий 100 ДляСупов	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ Морковка
ВВЕСТИ Мука белая сладкая 500 ДляПирожков	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ Мука
ВВЕСТИ Виноград красный кислый 10 ДляСалатов	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ Виноград Виноград красный кислый 10 ДляСалатов
СОРТИРОВАТЬ 3	Морковка оранжевый сладкий 100 ДляСупов
ИСКЛЮЧИТЬ Виноград	Мука белая сладкая 500 ДляПирожков
ОТОБРАЗИТЬ Морковка	ДАННЫЕ ОТСОРТИРОВАНЫ ПО 3
СОЗДАТЬ_ФАЙЛ kitchen.txt	ЗАПИСЬ Виноград УДАЛЕНА
ИСКЛЮЧИТЬ Мука	Морковка оранжевый сладкий 100
СОРТИРОВАТЬ 1	ДляСупов

ИМПОРТ kitchen.txt СОРТИРОВАТЬ 0	ИНФОРМАЦИЯ О ЗАПИСИ Морковка ПРЕДСТАВЛЕНА ВЫШЕ kitchen.txt ЗАПИСАН ЗАПИСЬ Мука УДАЛЕНА Морковка оранжевый сладкий 100 ДляСупов ДАННЫЕ ОТСОРТИРОВАНЫ ПО 1 kitchen.txt ЗАГРУЖЕН Морковка оранжевый сладкий 100 ДляСупов Мука белая сладкая 500 ДляПирожков ДАННЫЕ ОТСОРТИРОВАНЫ ПО 0
-------------------------------------	---

### 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 не корректно 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	ОТОБРАЗИТЬ	5 баллов	3 балла	0 баллов
3	ИСКЛЮЧИТЬ	5 баллов	3 балла	0 баллов
4	СОРТИРОВАТЬ	5 баллов	3 балла	0 баллов
5	СОЗДАТЬ ФАЙЛ	5 баллов	3 балла	0 баллов
6	ИМПОРТ	5 баллов	3 балла	0 баллов
<b>ИТОГО</b>		<b>30 баллов</b>	<b>18 баллов</b>	<b>0 баллов</b>

### Кейс №2.

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

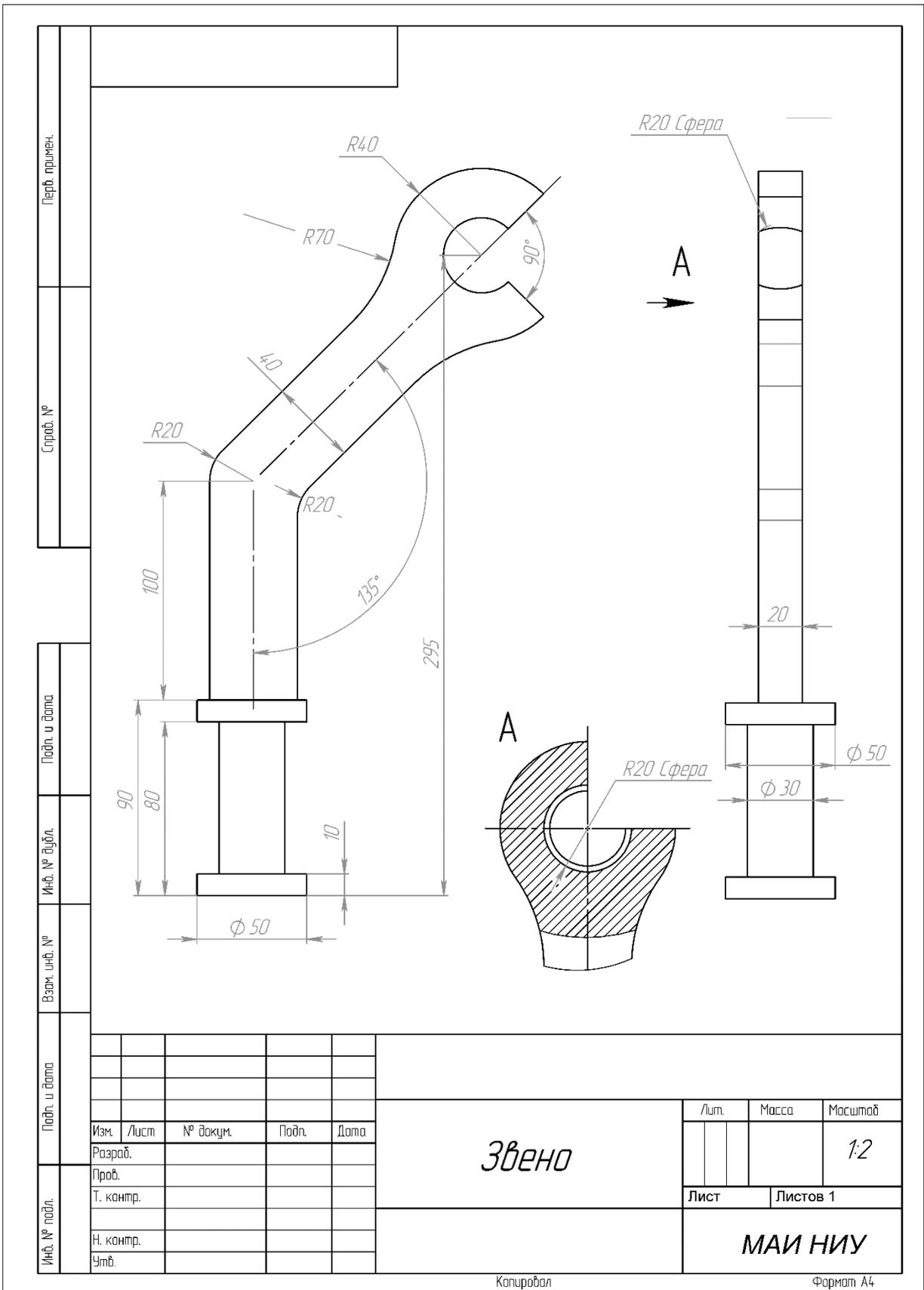
#### Задание 2

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.

### *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.





Перф. примен.	
Справ. №	
Подп. и дата	
Инд. № дубл.	
Взам. инд. №	
Подп. и дата	
Инд. № подл.	

Изм.	Лист	№ докум.	Подп.	Дата
Разраб.				
Проб.				
Т. контр.				
Н. контр.				
Утв.				

*Звено*

Лит.	Масса	Масштаб
		1:2
Лист		Листов 1
<b>МАИ НИУ</b>		

Копировал

Формат А4

Рисунок 2



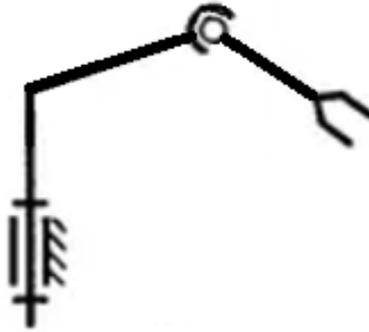


Рисунок 4

## Вариант 4

### Кейс №1.

«Программа учёта прохождения диспансеризации»

Разработать на языке Python программу с консольным интерфейсом для учёта результатов прохождения диспансеризации учеников. Данные о каждом ученике представляют собой совокупность следующих значений:

Номер	0	1	2	3	4
Поле	ФамилияИО	Возраст	Номер Школы	Специализация врача	Запись о результате осмотра

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:

1. **ВПИСАТЬ** <ФамилияИО> <Возраст> <Номер школы> <Специализация врача> <Запись о результате осмотра> – добавляет данные о новом ученике с заданными значениями. После выполнения функции в консоль приложения записывается фраза: **ДОБАВЛЕНА НОВАЯ ЗАПИСЬ** <Название>
2. **НАЙТИ** < ФамилияИО > – выводит в консоль всю информацию об ученике с заданной фамилией. После выполнения функции в консоль приложения записывается фраза: **ИНФОРМАЦИЯ О ЗАПИСИ** <Название> **ПРЕДСТАВЛЕНА ВЫШЕ**
3. **СТЕРЕТЬ** < ФамилияИО > – удаляет ученика с заданной фамилией. После выполнения функции в консоль приложения записывается фраза: **ЗАПИСЬ** <Название> **УДАЛЕНА**
4. **УПОРЯДОЧИТЬ** <номер\_поля\_сорт> – выводит в консоль информацию обо всех добавленных учениках, 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>.

5. ЭКСПОРТ <имя\_файла> – сохраняет добавленных учеников в заданный текстовый файл в формате команды ВПИСАТЬ (с разделением значений пробелами): <ФамилияИО> <Возраст> <Номер школы> <Специализация врача> <Запись о результате осмотра>. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> ЗАПИСАН.
6. СЧИТАТЬ <имя\_файла> – загружает данные об учениках из заданного текстового файла, полностью замещая текущий состав записей об учениках. Считается, что загружаемый файл был создан при вызове команды ЭКСПОРТ. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> ЗАГРУЖЕН.

Описание параметров команд:

- <ФамилияИО> – фамилия ученика, БЕЗ разделения пробелами (поле №0), например, КарповДИ. Данное значение является уникальным для каждого результата.
- <Возраст> – классификация ученика, натуральное число (поле №1).
- <Номер Школы> – классификация ученика, натуральное число (поле №2).
- <Специализация врача> – врач, проводивший осмотр ученику, строковое значение, не содержащее пробелов (поле №3).
- <Запись о результате осмотра> – текстовая заметка об осмотре, 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
ВПИСАТЬ ЗапорожскихАИ 15 2 Дерматолог Здоров	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ ЗапорожскихАИ
ВПИСАТЬ БеловаНИ 17 5 Окулист	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ БеловаНИ
ПрийтиНаПовторное	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ ИвановАБ
ВПИСАТЬ ИвановАБ 13 12 Терапевт Здоров	ИвановАБ 13 12 Терапевт Здоров ЗапорожскихАИ 15 2 Дерматолог Здоров
УПОРЯДОЧИТЬ 1	БеловаНИ 17 5 Окулист ПрийтиНаПовторное
СТЕРЕТЬ ИвановАБ	ДАННЫЕ ОТСОРТИРОВАНЫ ПО 1
НАЙТИ БеловаНИ	ЗАПИСЬ ИвановАБ УДАЛЕНА
ЭКСПОРТ pupils.txt	БеловаНИ 17 5 Окулист ПрийтиНаПовторное
СТЕРЕТЬ БеловаНИ	ИНФОРМАЦИЯ О ЗАПИСИ БеловаНИ
УПОРЯДОЧИТЬ 2	ПРЕДСТАВЛЕНА ВЫШЕ
СЧИТАТЬ pupils.txt	pupils.txt ЗАПИСАН
УПОРЯДОЧИТЬ 0	ЗАПИСЬ БеловаНИ УДАЛЕНА ЗапорожскихАИ 15 2 Дерматолог Здоров ДАННЫЕ ОТСОРТИРОВАНЫ ПО 2

	<p> pupils.txt ЗАГРУЖЕН  БеловаНИ 17 5 Окулист ПрийтиНаПовторное  ЗапорожскихАИ 15 2 Дерматолог Здоров  ДАННЫЕ ОТСОРТИРОВАНЫ ПО 0</p>
--	---

### 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 не корректно 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	ВПИСАТЬ	5 баллов	3 балла	0 баллов
2	НАЙТИ	5 баллов	3 балла	0 баллов
3	СТЕРЕТЬ	5 баллов	3 балла	0 баллов
4	УПОРЯДОЧИТЬ	5 баллов	3 балла	0 баллов
5	ЭКСПОРТ	5 баллов	3 балла	0 баллов
6	СЧИТАТЬ	5 баллов	3 балла	0 баллов
<b>ИТОГО</b>		<b>30 баллов</b>	<b>18 баллов</b>	<b>0 баллов</b>

### Кейс №2.

#### Задание 1

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

#### Задание 2

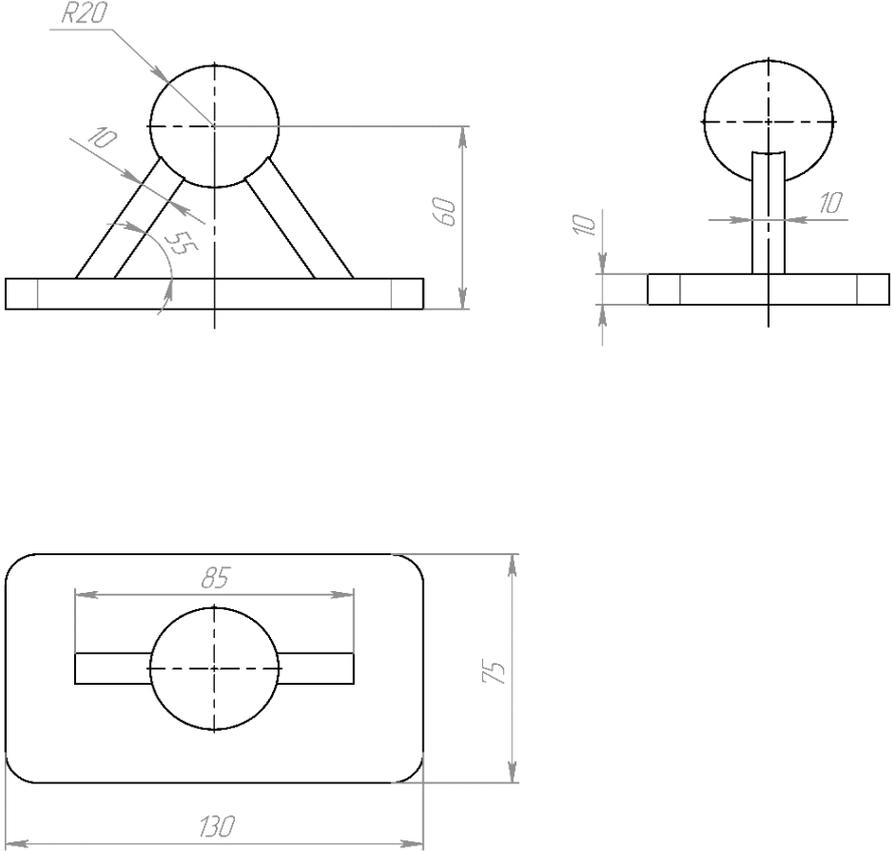
Соберите детали из 1ого задания в сборку согласно кинематической схеме, приведенной на рисунке 4. Пересечения деталей не допускается. Экпортируйте сборку в формат .stl. – 10 баллов.

#### 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. Экпортируйте сборку в формат .stl. – 5 баллов.

Перб. примен.					
Стр. №					
Подп. и дата					
Инб. № дубл.					
Взам. инб. №					
Подп. и дата					
Инб. № подл.					

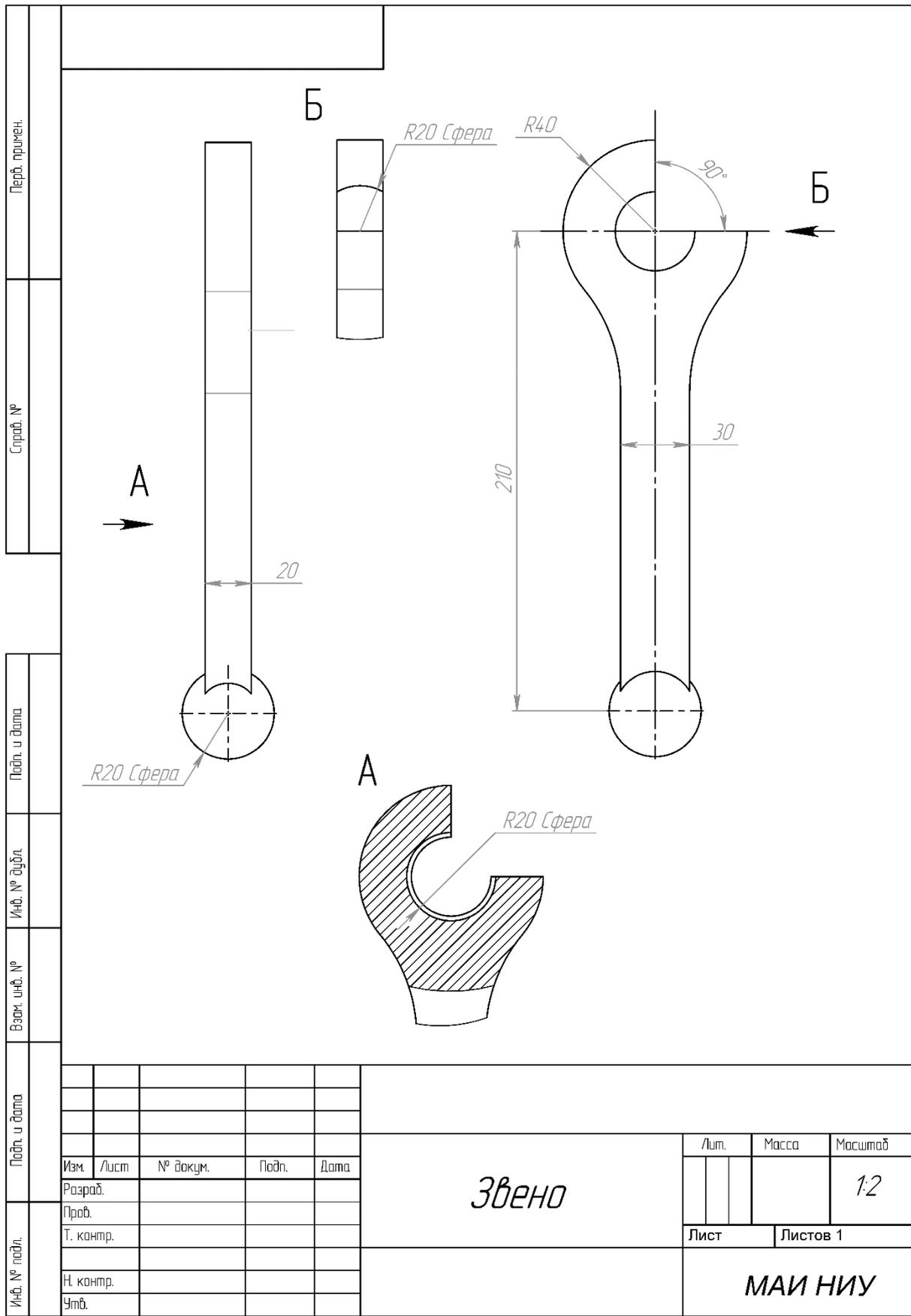


						<i>Основание</i>									
Изм.	Лист	№ докум.	Подп.	Дата			Лит.	Масса	Масштаб						
Разраб.									1:2						
Проб.															
Т. контр.							Лист	Листов 1							
Н. контр.							<i>МАИ НИУ</i>								
Чтв.															

Копировал

Формат А4

Рисунок 1



Перв. примен.	Спраб. №	Подп. и дата	Инд. № дубл.	Взам. инд. №	Подп. и дата	Инд. № подл.

Изм.	Лист	№ докум.	Подп.	Дата
Разраб.				
Проф.				
Т. контр.				
Н. контр.				
Утв.				

**Звено**

Лит.	Масса	Масштаб
		1:2
Лист	Листов 1	
<b>МАИ НИУ</b>		

Копировал

Формат А4

Рисунок 2



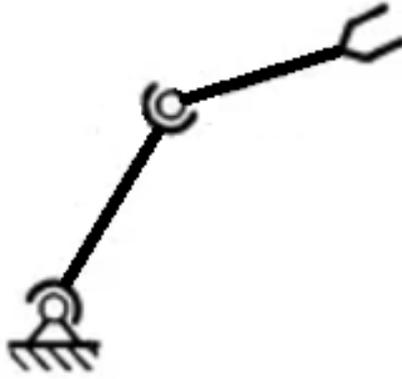


Рисунок 4

## Вариант 5

### Кейс №1.

«Программа учёта природных ресурсов государств»

Разработать на языке Python программу с консольным интерфейсом для учёта природных ресурсов государств. Данные о каждом природном ресурсе представляют собой совокупность следующих значений:

Номер	0	1	2	3	4
Поле	Название	Страна	Вид ресурса	Запас в триллионах тонн	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:

1. **ЗАДАТЬ** <Название> <Страна> <Вид ресурса> <Запаса в триллионах тонн> <Текстовая заметка> – добавляет данные о новом природном ресурсе с заданными значениями. После выполнения функции в консоль приложения записывается фраза: **ДОБАВЛЕНА НОВАЯ ЗАПИСЬ** <Название>
2. **ВЫВЕСТИ** <Название> – выводит в консоль всю информацию о природном ресурсе с заданным названием. После выполнения функции в консоль приложения записывается фраза: **ИНФОРМАЦИЯ О ЗАПИСИ** <Название> **ПРЕДСТАВЛЕНА ВЫШЕ**
3. **УБРАТЬ** <Название> – удаляет природный ресурс с заданным названием. После выполнения функции в консоль приложения записывается фраза: **ЗАПИСЬ** <Название> **УДАЛЕНА**
4. **СПИСОК** <номер\_поля\_сорт> – выводит в консоль информацию обо всех добавленных природных ресурсах, 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>.

5. ЗАПИСАТЬ <имя\_файла> – сохраняет добавленные природные ресурсы в заданный текстовый файл в формате команды ЗАДАТЬ (с разделением значений пробелами): <Название> <Страна> <Вид ресурса> <Запас в триллионах тонн> <Текстовая заметка>. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> ЗАПИСАН
6. ПРОЧИТАТЬ\_ФАЙЛ <имя\_файла> – загружает данные о природных ресурсах из заданного текстового файла, полностью замещая текущий состав записей о природных ресурсах. 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 природного ресурса, БЕЗ разделения пробелами (поле №0), например, ЖелезнаяРуда. Данное значение является уникальным для каждого ресурса.
- <Страна> – место добычи природного ресурса, строковое значение, не содержащее пробелов (поле №1).
- <Вид ресурса> – классификация природного ресурса, строковое значение, не содержащее пробелов (поле №2).
- <Запас в триллионах тонн> – количество природного ресурса, натуральное число (поле №3).
- <Текстовая заметка> – текстовая заметка о природном ресурсе, 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
ЗАДАТЬ Леса Казахстан невозобновляемые 90 Смолистые	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ Леса
ЗАДАТЬ Вода Китай возобновляемые 4035 НельзяПить	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ Вода
ЗАДАТЬ Уголь Россия невозобновляемые 1253 Дешевый	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ Уголь
СПИСОК 3	Леса Казахстан невозобновляемые 90
УБРАТЬ Вода	Смолистые
ВЫВЕСТИ Леса	Уголь Россия невозобновляемые 1253
ЗАПИСАТЬ resources.txt	Дешевый
УБРАТЬ Леса	Вода Китай возобновляемые 4035 НельзяПить
СПИСОК 2	ДАННЫЕ ОТСОРТИРОВАНЫ ПО 3
ПРОЧИТАТЬ_ФАЙЛ resources.txt	ЗАПИСЬ Вода УДАЛЕНА
СПИСОК 0	Леса Казахстан невозобновляемые 90
	Смолистые
	ИНФОРМАЦИЯ О ЗАПИСИ Леса
	ПРЕДСТАВЛЕНА ВЫШЕ

	resources.txt ЗАПИСАН ЗАПИСЬ Леса УДАЛЕНА Уголь Россия невозобновляемые 1253 Дешевый ДАННЫЕ ОТСОРТИРОВАНЫ ПО 2 resources.txt ЗАГРУЖЕН Леса Казахстан невозобновляемые 90 Смолистые Уголь Россия невозобновляемые 1253 Дешевый ДАННЫЕ ОТСОРТИРОВАНЫ ПО 0
--	---

### 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 не корректно 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	ЗАДАТЬ	5 баллов	3 балла	0 баллов
2	ВЫВЕСТИ	5 баллов	3 балла	0 баллов
3	УБРАТЬ	5 баллов	3 балла	0 баллов
4	СПИСОК	5 баллов	3 балла	0 баллов
5	ЗАПИСАТЬ	5 баллов	3 балла	0 баллов
6	ПРОЧИТАТЬ ФАЙЛ	5 баллов	3 балла	0 баллов
<b>ИТОГО</b>		<b>30 баллов</b>	<b>18 баллов</b>	<b>0 баллов</b>

## Кейс №2.

### Задание 1

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

### Задание 2

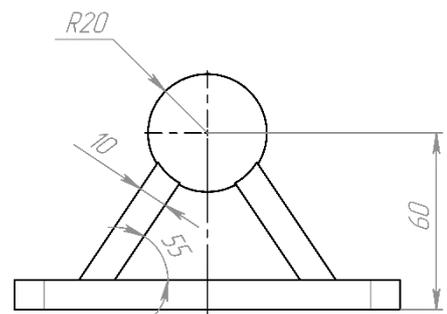
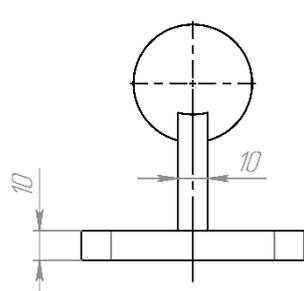
Соберите детали из 1ого задания в сборку согласно кинематической схеме, приведенной на рисунке 4. Пересечения деталей не допускается. Экпортируйте сборку в формат .stl. – 10 баллов.

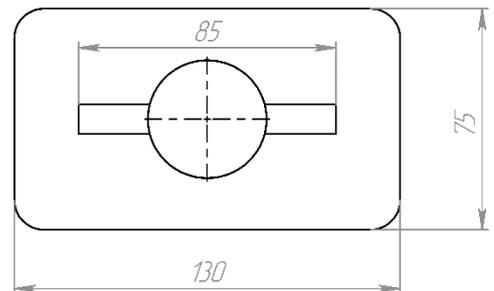
### Задание 3

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.

Перв. примен.						
Спроб. №						
Подп. и дата						
Инв. № дубл.						
Взам. инв. №						
Подп. и дата						
Инв. № подл.						
Изм.	Лист	№ докум.	Подп.	Дата	<i>Основание</i>	
Разраб.						
Проб.						
Т. контр.						
Н. контр.						
Утв.					Лист	Листов 1
					<i>МАИ НИУ</i>	



Копировал

Формат А4

Рисунок 1

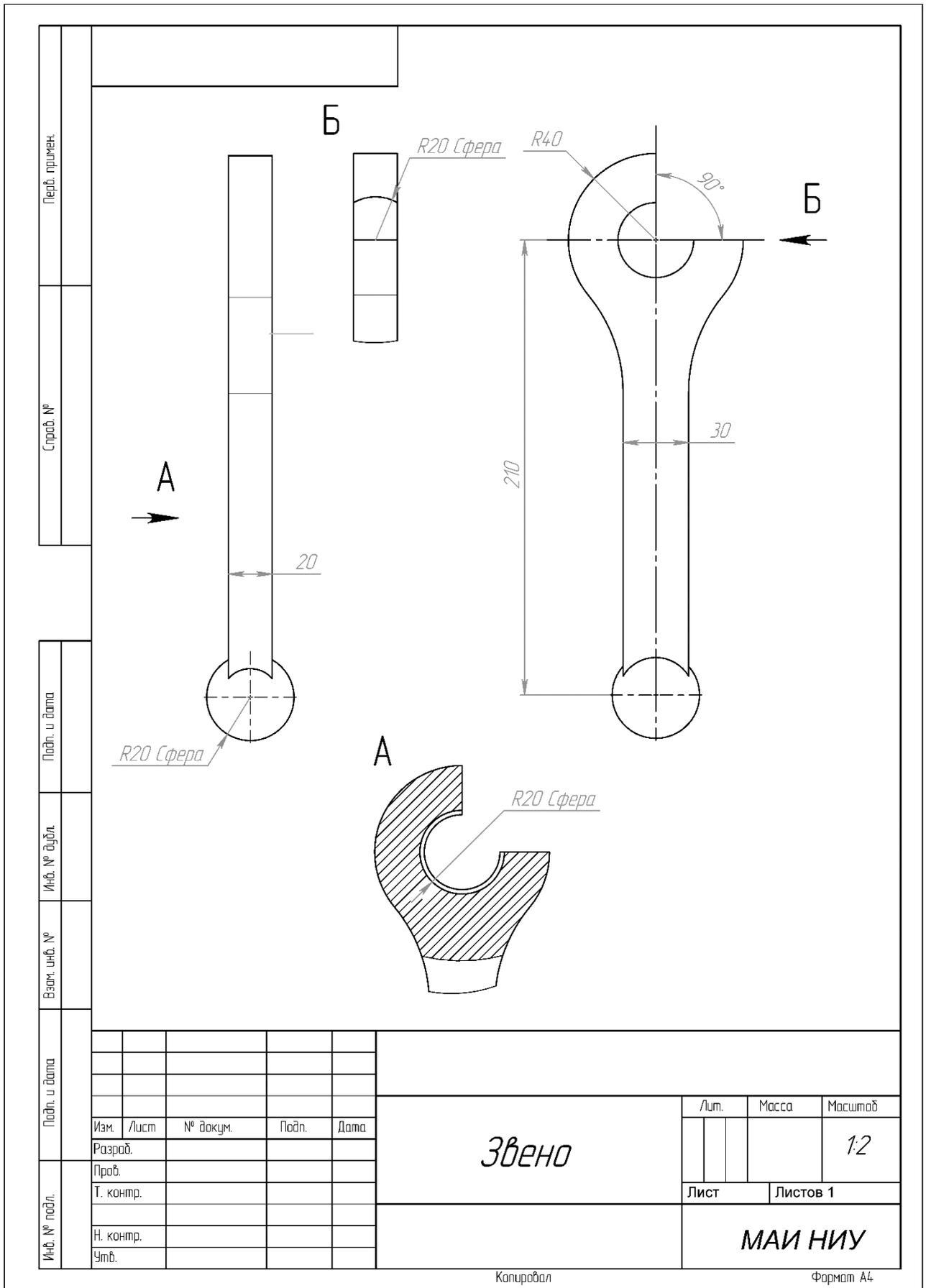


Рисунок 2

Перв. примен.																																																																																	
Спроб. №																																																																																	
Подп. и дата																																																																																	
Инв. № дубл.																																																																																	
Взам. инв. №	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;"></td> </tr> <tr> <td>Изм.</td> <td>Лист</td> <td>№ докум.</td> <td>Подп.</td> <td>Дата</td> <td colspan="5" rowspan="5" style="text-align: center; vertical-align: middle; font-size: 2em;"><i>Звено</i></td> <td>Лит.</td> <td>Масса</td> <td>Масштаб</td> </tr> <tr> <td>Разраб.</td> <td></td> <td>1:5</td> </tr> <tr> <td>Проб.</td> <td></td> <td></td> <td></td> <td></td> <td colspan="2">Лист</td> <td colspan="2">Листов 1</td> </tr> <tr> <td>Т. контр.</td> <td></td> <td></td> <td></td> <td></td> <td colspan="6" style="text-align: center; font-weight: bold; font-size: 1.2em;">МАИ НИУ</td> </tr> <tr> <td>Н. контр.</td> <td></td> <td></td> <td></td> <td></td> <td colspan="6"></td> </tr> <tr> <td>Утв.</td> <td></td> <td></td> <td></td> <td></td> <td colspan="6"></td> </tr> </table>														Изм.	Лист	№ докум.	Подп.	Дата	<i>Звено</i>					Лит.	Масса	Масштаб	Разраб.											1:5	Проб.					Лист		Листов 1		Т. контр.					МАИ НИУ						Н. контр.											Утв.										
Изм.	Лист	№ докум.	Подп.	Дата	<i>Звено</i>					Лит.	Масса	Масштаб																																																																					
Разраб.																1:5																																																																	
Проб.										Лист		Листов 1																																																																					
Т. контр.										МАИ НИУ																																																																							
Н. контр.																																																																																	
Утв.																																																																																	
Подп. и дата																																																																																	
Инв. № подл.																																																																																	

Копировал

Формат А4

Рисунок 3

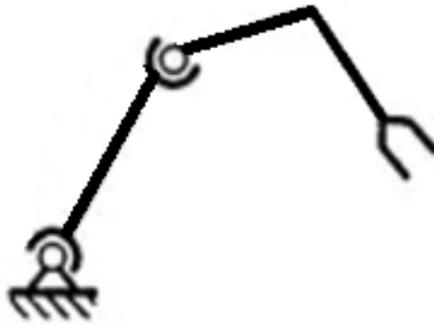


Рисунок 4

## Вариант 6

### Кейс №1.

«Программа учёта автомобилей в автосалоне»

Разработать на языке Python программу с консольным интерфейсом для учёта автомобилей. Данные о каждом автомобиле представляют собой совокупность следующих значений:

Номер	0	1	2	3	4
Поле	ФамилияИО	Марка	Цвет	Стоимость	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:

1. **ВВЕСТИ** <ФамилияИО> <Марка> <Цвет> <Стоимость> <Текстовая заметка> – добавляет данные о новом автомобиле с заданными значениями. После выполнения функции в консоль приложения записывается фраза: **ДОБАВЛЕНА НОВАЯ ЗАПИСЬ** <Название>
2. **ВЫВЕСТИ** <ФамилияИО> – выводит в консоль всю информацию об автомобиле с заданной фамилией владельца. После выполнения функции в консоль приложения записывается фраза: **ИНФОРМАЦИЯ О ЗАПИСИ** <Название> **ПРЕДСТАВЛЕНА ВЫШЕ**
3. **СТЕРЕТЬ** <ФамилияИО> – удаляет автомобиль с заданной фамилией владельца. После выполнения функции в консоль приложения записывается фраза: **ЗАПИСЬ** <Название> **УДАЛЕНА**
4. **СОРТИРОВАТЬ** <номер\_поля\_сорт> – выводит в консоль информацию обо всех добавленных автомобилях, 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>.

5. ЗАПИСАТЬ <имя\_файла> – сохраняет добавленные автомобили в заданный текстовый файл в формате команды ВВЕСТИ (с разделением значений пробелами): <ФамилияИО> <Марка> <Цвет> <Стоимость> <Текстовая заметка>. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> ЗАПИСАН.
6. СЧИТАТЬ <имя\_файла> – загружает данные об автомобиле из заданного текстового файла, полностью заменяя текущий состав записей об автомобилях. Считается, что загружаемый файл был создан при вызове команды ЗАПИСАТЬ. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> ЗАГРУЖЕН.

Описание параметров команд:

- <ФамилияИО> – фамилия владельца автомобиля, БЕЗ разделения пробелами (поле №0), например, АкатьевЯВ. Данное значение является уникальным для каждого автомобиля.
- <Марка> – классификация автомобиля, строковое значение, не содержащее пробелов (поле №1).
- <Цвет> – характеристика автомобиля, строковое значение, не содержащее пробелов (поле №2).
- <Стоимость> – характеристика автомобиля, натуральное число (поле №3).
- <Текстовая заметка> – текстовая заметка об автомобиле, 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 АлишерБГ УАЗ Зеленый 1000000 ОтечАвтоПром	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ АлишерБГ ДОБАВЛЕНА НОВАЯ ЗАПИСЬ НиколаевЛД
ВВЕСТИ НиколаевЛД КАМАЗ Оранжевый 3240000 ОтечАвтоПром	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ УженковДИ АлишерБГ УАЗ Зеленый 1000000
ВВЕСТИ УженковДИ Рено белая 1433944 Битая	ОтечАвтоПром УженковДИ Рено белая 1433944 Битая
СОРТИРОВАТЬ 3	НиколаевЛД КАМАЗ Оранжевый 3240000
СТЕРЕТЬ АлишерБГ	ОтечАвтоПром
ВЫВЕСТИ УженковДИ	ДАННЫЕ ОТСОРТИРОВАНЫ ПО 3
ЗАПИСАТЬ auto.txt	ЗАПИСЬ АлишерБГ УДАЛЕНА
СТЕРЕТЬ НиколаевЛД	УженковДИ Рено белая 1433944 Битая
СОРТИРОВАТЬ 2	ИНФОРМАЦИЯ О ЗАПИСИ
СЧИТАТЬ auto.txt	УженковДИ ПРЕДСТАВЛЕНА ВЫШЕ
СОРТИРОВАТЬ 0	auto.txt ЗАПИСАН

	ЗАПИСЬ НиколаевЛД УДАЛЕНА УженковДИ Рено белая 1433944 Битая ДАННЫЕ ОТСОРТИРОВАНЫ ПО 2 auto.txt ЗАГРУЖЕН УженковДИ Рено белая 1433944 Битая НиколаевЛД КАМАЗ Оранжевый 3240000 ОтечАвтоПром ДАННЫЕ ОТСОРТИРОВАНЫ ПО 0
--	--

### 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 не корректно 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
<b>ИТОГО</b>		<b>30 баллов</b>	<b>18 баллов</b>	<b>0 баллов</b>

## Кейс №2.

### Задание 1

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

### 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.

### 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.

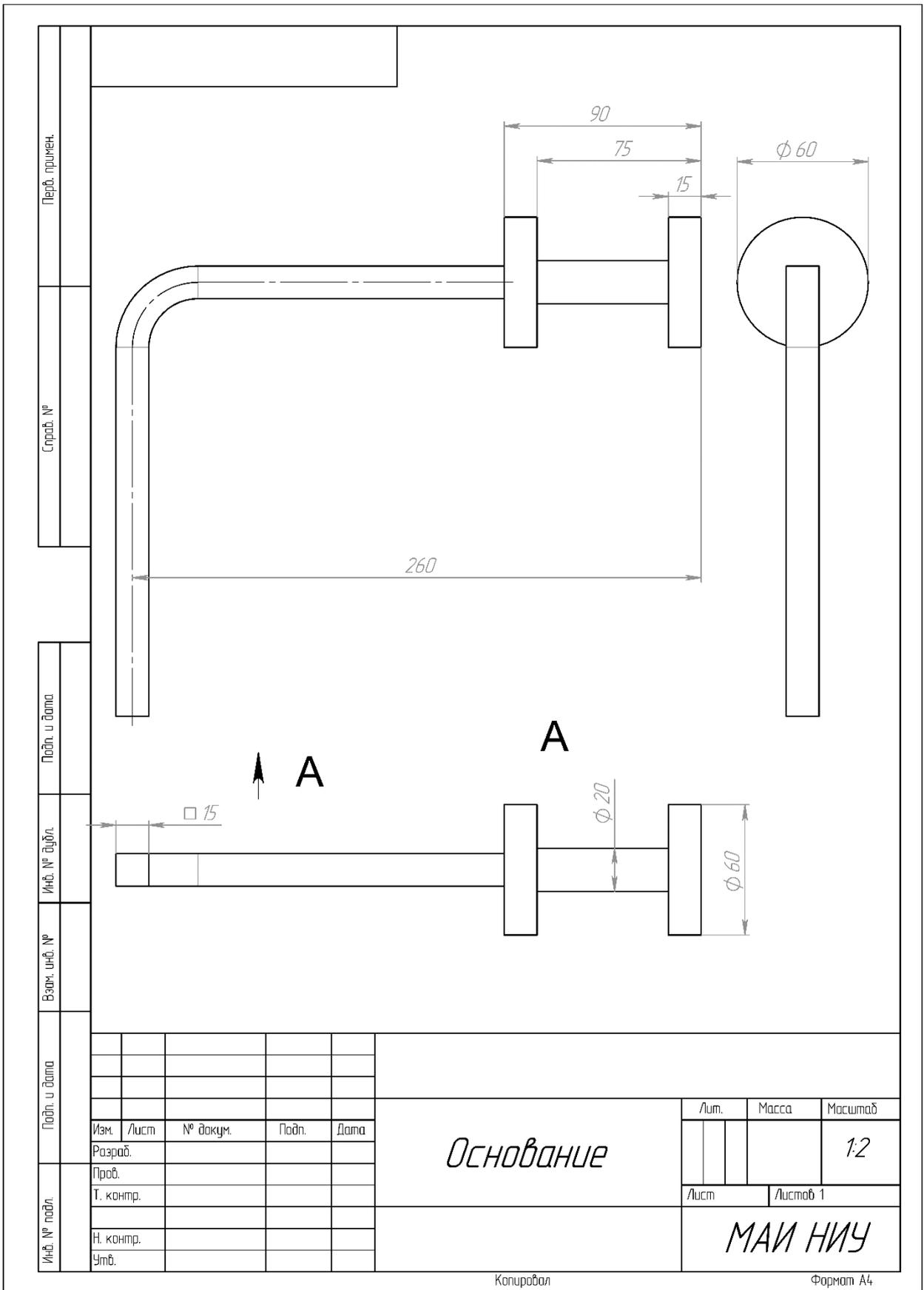


Рисунок 1

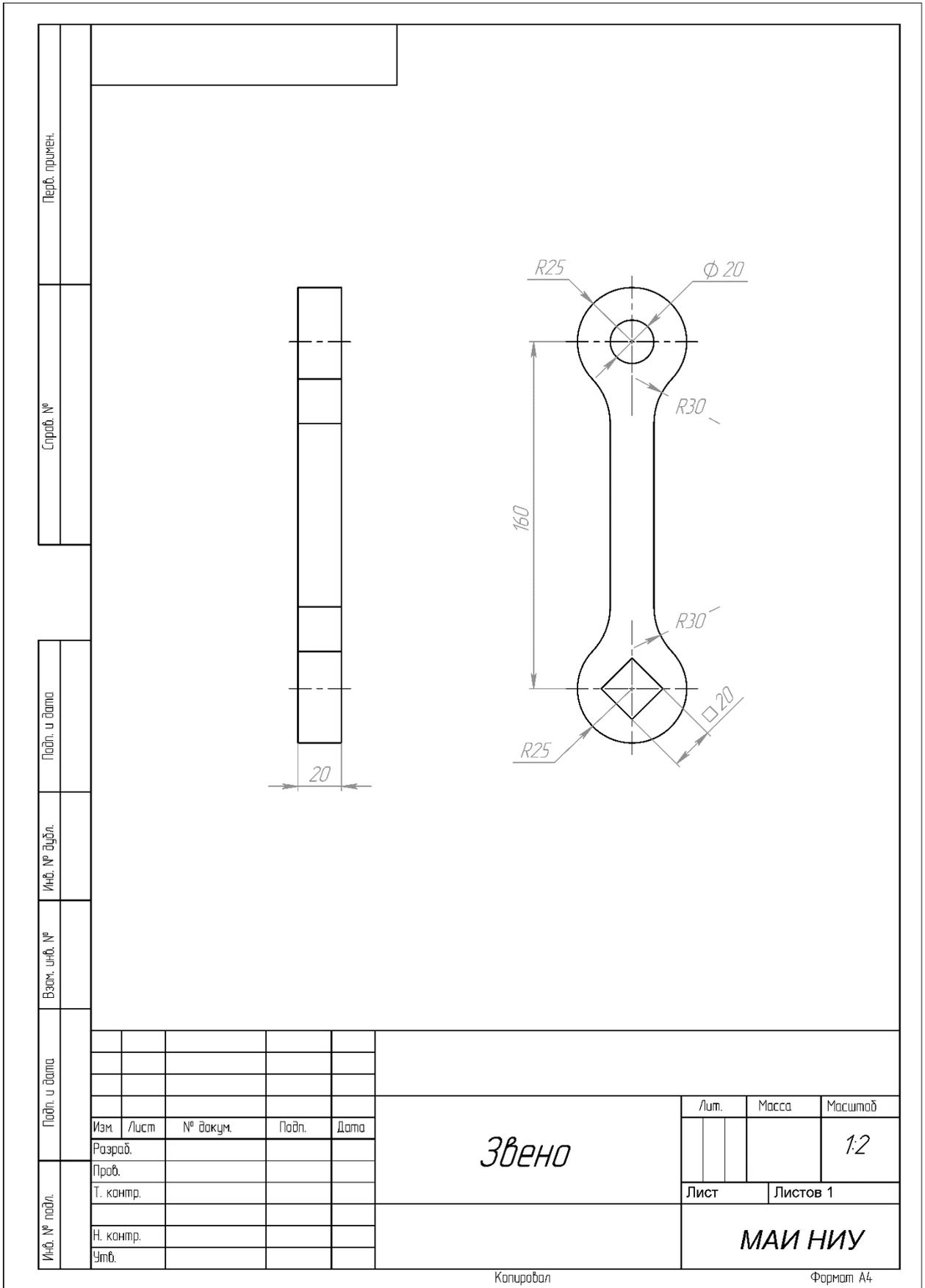


Рисунок 2



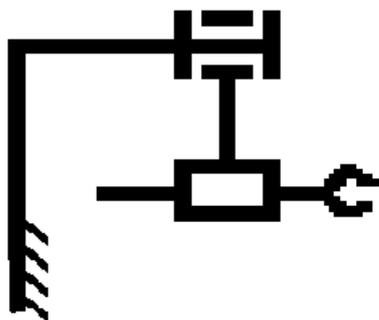


Рисунок 4

## Вариант 7

### Кейс №1.

«Программа учёта блюд для планирования питания»

Разработать на языке Python программу с консольным интерфейсом для учёта блюд. Данные о каждом блюде представляют собой совокупность следующих значений:

Номер	0	1	2	3	4
Поле	Название	Питательное вещество	Содержание питательного вещества в граммах	Витамин	Содержание витамина в граммах

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:

1. **ЗАДАТЬ** <Название> <Питательное вещество> <Содержание питательного вещества в граммах> <Витамин> <Содержание витамина в граммах> – добавляет данные о новом блюде с заданными значениями. После выполнения функции в консоль приложения записывается фраза: **ДОБАВЛЕНА НОВАЯ ЗАПИСЬ** <Название>
2. **НАЙТИ** <Название> – выводит в консоль всю информацию о блюде с заданным названием. После выполнения функции в консоль приложения записывается фраза: **ИНФОРМАЦИЯ О ЗАПИСИ** <Название> **ПРЕДСТАВЛЕНА ВЫШЕ**
3. **ИСКЛЮЧИТЬ** <Название> – удаляет блюдо с заданным названием. После выполнения функции в консоль приложения записывается фраза: **ЗАПИСЬ** <Название> **УДАЛЕНА**
4. **СПИСОК** <номер\_поля\_сорт> – выводит в консоль информацию обо всех добавленных блюдах, 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>.

5. ЭКСПОРТ <имя\_файла> – сохраняет добавленные блюда в заданный текстовый файл в формате команды ЗАДАТЬ (с разделением значений пробелами): <Название> <Питательное вещество> <Содержание питательного вещества в граммах> <Витамин> <Содержание витамина в граммах>. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> ЗАПИСАН.
6. ИМПОРТ <имя\_файла> – загружает данные о блюдах из заданного текстового файла, полностью замещая текущий состав записей о блюдах. Считается, что загружаемый файл был создан при вызове команды ЭКСПОРТ. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> ЗАГРУЖЕН.

Описание параметров команд:

- <Название> – название блюда, БЕЗ разделения пробелами (поле №0), например, КислыеЩи. Данное значение является уникальным для каждого блюда.
- <Питательное вещество> – классификация блюда, строковое значение, не содержащее пробелов (поле №1).
- <Содержание питательного вещества в граммах> – количество питательного вещества в блюде, натуральное число (поле №2).
- <Витамин> – характеристика блюда, строковое значение, не содержащее пробелов (поле №3).
- <Содержание витамина в граммах> – количество витамина в блюде, вещественное число (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
ЗАДАТЬ ВареныйШпинат Клетчатка 10 С 0.1	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ ВареныйШпинат
ЗАДАТЬ КартофельноеПюре Крахмал 80 В 0.3	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ КартофельноеПюре
ЗАДАТЬ ХариусГорячегоКопчения Белок 25 Omega3 5.0	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ ХариусГорячегоКопчения
СПИСОК 2	ВареныйШпинат Клетчатка 10 С 0.1 ХариусГорячегоКопчения Белок 25 Omega3 5.0
ИСКЛЮЧИТЬ ХариусГорячегоКопчения	КартофельноеПюре Крахмал 80 В 0.3
НАЙТИ КартофельноеПюре	ДАННЫЕ ОТСОРТИРОВАНЫ ПО 2
ЭКСПОРТ dishes.txt	ЗАПИСЬ ХариусГорячегоКопчения УДАЛЕНА
ИСКЛЮЧИТЬ КартофельноеПюре	КартофельноеПюре Крахмал 80 В 0.3
СПИСОК 1	ИНФОРМАЦИЯ О ЗАПИСИ
ИМПОРТ dishes.txt	КартофельноеПюре ПРЕДСТАВЛЕНА ВЫШЕ dishes.txt ЗАПИСАН
СПИСОК 0	ЗАПИСЬ КартофельноеПюре УДАЛЕНА ВареныйШпинат Клетчатка 10 С 0.1 ДАННЫЕ ОТСОРТИРОВАНЫ ПО 1 dishes.txt ЗАГРУЖЕН

	ВареныйШпинат Клетчатка 10 С 0.1 КартофельноеПюре Крахмал 80 В 0.3 <b>ДАННЫЕ ОТСОРТИРОВАНЫ ПО 0</b>
--	---

### 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 не корректно 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	ЗАДАТЬ	5 баллов	3 балла	0 баллов
2	НАЙТИ	5 баллов	3 балла	0 баллов
3	ИСКЛЮЧИТЬ	5 баллов	3 балла	0 баллов
4	СПИСОК	5 баллов	3 балла	0 баллов
5	ЭКСПОРТ	5 баллов	3 балла	0 баллов
6	ИМПОРТ	5 баллов	3 балла	0 баллов
<b>ИТОГО</b>		<b>30 баллов</b>	<b>18 баллов</b>	<b>0 баллов</b>

### Кейс №2.

#### Задание 1

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

#### Задание 2

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

#### 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. Экпортируйте сборку в формат .stl. – 5 баллов.

Перв. примен.				
Справ. №				
Подл. и дата				
Инв. № дубл.				
Взам. инв. №				
Подл. и дата				
Инв. № подл.				

Изм.	Лист	№ докум.	Подп.	Дата	<h2>Основание</h2>					Лит.	Масса	Масштаб
Разраб.										1:2		
Проб.										Лист	Листов 1	
Т. контр.										<h2>МАИ НИУ</h2>		
Н. контр.												
Утв.												

Копировал

Формат А4

Рисунок 1

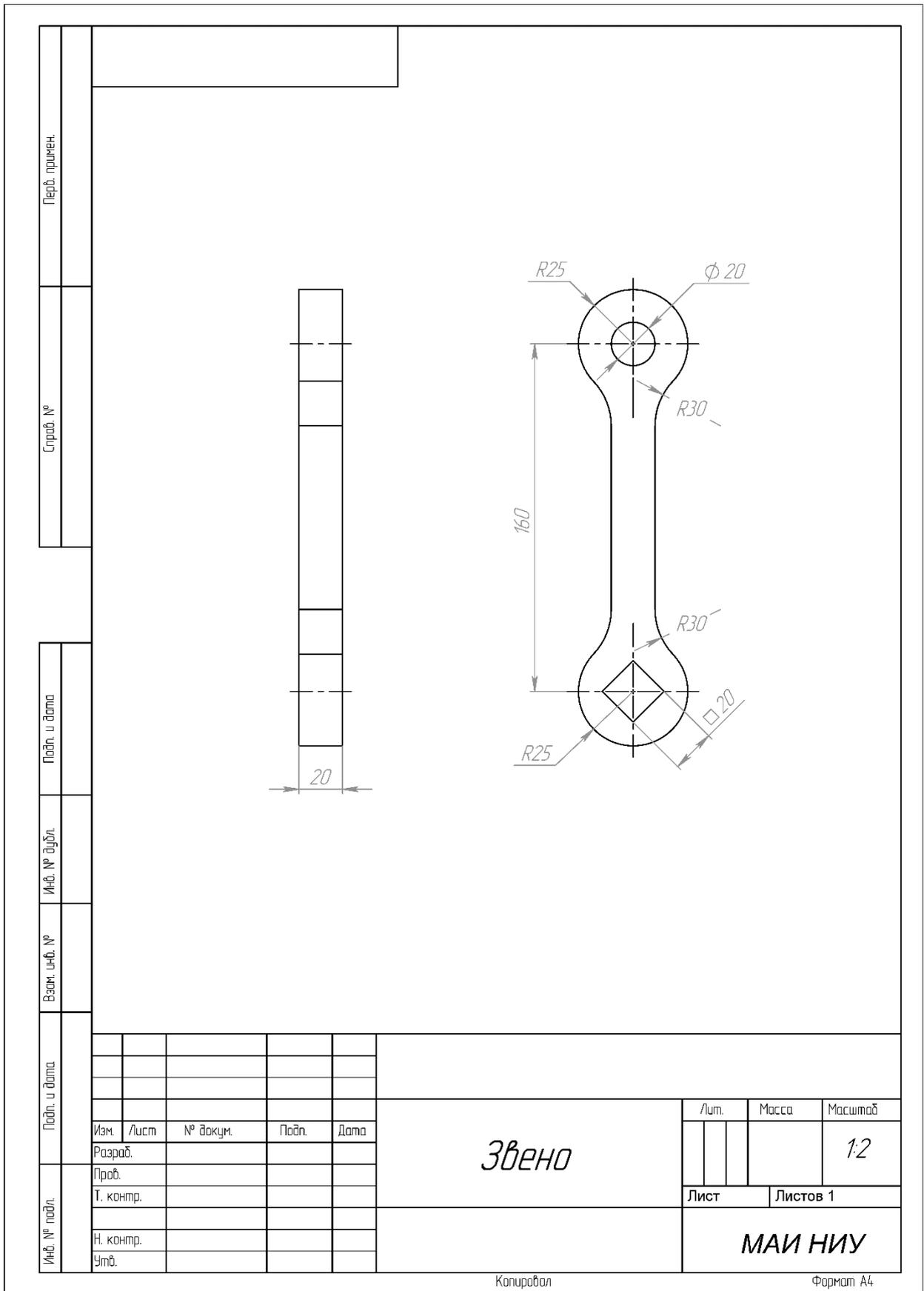


Рисунок 2

Перв. примен.								
Спроб. №								
Подп. и дата								
Инв. № дубл.								
Взам. инв. №								
Подп. и дата								
Инв. № подл.								
Т. контр.								
Н. контр.								
Утв.								

*Звено*

Лит.	Масса	Масштаб
		1:2
Лист		Листов 1
МАИ НИУ		

Копировал

Формат А4

Рисунок 3

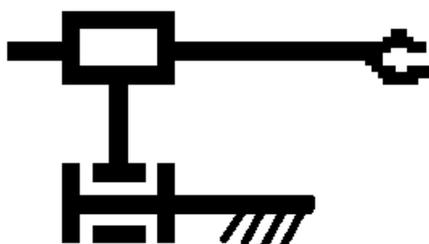


Рисунок 4

## Вариант 8

### Кейс №1.

«Программа учёта космических тел»

Разработать на языке Python программу с консольным интерфейсом для учёта космических тел. Данные о каждом космическом теле представляют собой совокупность следующих значений:

Номер	0	1	2	3	4
Поле	Название	Вид	Предполагаемая масса (мантисса*)	Предполагаемая масса (Степень десятки*)	Текстовая запись

\*Предполагаемая масса складывается из двух частей из <мантиссы>, умноженной на 10 в степени из колонки <степень десятки>

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:

1. **ВПИСАТЬ** <Название> <Вид> <Предполагаемая масса (мантисса)> <Предполагаемая масса (Степень десятки)> <Текстовая запись> – добавляет данные о новом космическом теле с заданными значениями. После выполнения функции в консоль приложения записывается фраза: **ДОБАВЛЕНА НОВАЯ ЗАПИСЬ** <Название>
2. **НАЙТИ** <Название> – выводит в консоль всю информацию о космическом теле с заданным названием. После выполнения функции в консоль приложения записывается фраза: **ИНФОРМАЦИЯ О ЗАПИСИ** <Название> **ПРЕДСТАВЛЕНА ВЫШЕ**
3. **СТЕРЕТЬ** <Название> – удаляет космическое тело с заданным названием. После выполнения функции в консоль приложения записывается фраза: **ЗАПИСЬ** <Название> **УДАЛЕНА**
4. **УПОРЯДОЧИТЬ** <номер\_поля\_сорт> – выводит в консоль информацию обо всех добавленных космических телах, 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>.

5. ЭКСПОРТ <имя\_файла> – сохраняет добавленные космические тела в заданный текстовый файл в формате команды ВПИСАТЬ (с разделением значений пробелами): <Название> <Вид> <Предполагаемая масса (мантисса)> <Предполагаемая масса (Степень десятки)> <Текстовая запись>. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> ЗАПИСАН.
6. СЧИТАТЬ <имя\_файла> – загружает данные о космических телах из заданного текстового файла, полностью замещая текущий состав записей о космических телах. Считается, что загружаемый файл был создан при вызове команды ЭКСПОРТ. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> ЗАГРУЖЕН.

Описание параметров команд:

- <Название> – название космического тела, БЕЗ разделения пробелами (поле №0), например, ТретьяПланета. Данное значение является уникальным для каждого космического тела.
- <Вид> – классификация космического тела, строковое значение, не содержащее пробелов (поле №1).
- <Предполагаемая масса (мантисса)> – характеристика массы космического тела, положительное вещественное число (поле №2).
- <Предполагаемая масса (Степень десятки)> – характеристика массы космического тела, натуральное число (поле №3).
- <Текстовая запись> – текстовая заметка о космическом теле, 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
ВПИСАТЬ Астероид1313 комета 4.3 10 Безопасен	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ Астероид1313
ВПИСАТЬ Kepler5432 планета 7.3 210 ВозможнаЖизнь	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ Kepler5432
ВПИСАТЬ Солнце звезда 2.0 270 БлижайшаяЗвезда	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ Солнце
УПОРЯДОЧИТЬ 3	Астероид1313 комета 4.3 10 Безопасен
СТЕРЕТЬ Солнце	Kepler5432 планета 7.3 210 ВозможнаЖизнь
НАЙТИ Астероид1313	Солнце звезда 0.2 270 БлижайшаяЗвезда
ЭКСПОРТ space.txt	ДАННЫЕ ОТСОРТИРОВАНЫ ПО 3
СТЕРЕТЬ Kepler5432	ЗАПИСЬ Солнце УДАЛЕНА
УПОРЯДОЧИТЬ 1	Астероид1313 комета 4.3 10 Безопасен
	ИНФОРМАЦИЯ О ЗАПИСИ Астероид1313
	ПРЕДСТАВЛЕНА ВЫШЕ
	space.txt ЗАПИСАН

СЧИТАТЬ space.txt УПОРЯДОЧИТЬ 0	ЗАПИСЬ Kepler5432 УДАЛЕНА Астероид1313 комета 4.3 10 Безопасен ДАННЫЕ ОТСОРТИРОВАНЫ ПО 1 space.txt ЗАГРУЖЕН Kepler5432 планета 7.3 210 ВозможнаЖизнь Астероид1313 комета 4.3 10 Безопасен ДАННЫЕ ОТСОРТИРОВАНЫ ПО 0
------------------------------------	---

### 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 не корректно 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	ВПИСАТЬ	5 баллов	3 балла	0 баллов
2	НАЙТИ	5 баллов	3 балла	0 баллов
3	СТЕРЕТЬ	5 баллов	3 балла	0 баллов
4	УПОРЯДОЧИТЬ	5 баллов	3 балла	0 баллов
5	ЭКСПОРТ	5 баллов	3 балла	0 баллов
6	СЧИТАТЬ	5 баллов	3 балла	0 баллов
<b>ИТОГО</b>		<b>30 баллов</b>	<b>18 баллов</b>	<b>0 баллов</b>

## Кейс №2.

### Задание 1

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

### Задание 2

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

### 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. Экпортируйте сборку в формат .stl. – 5 баллов.

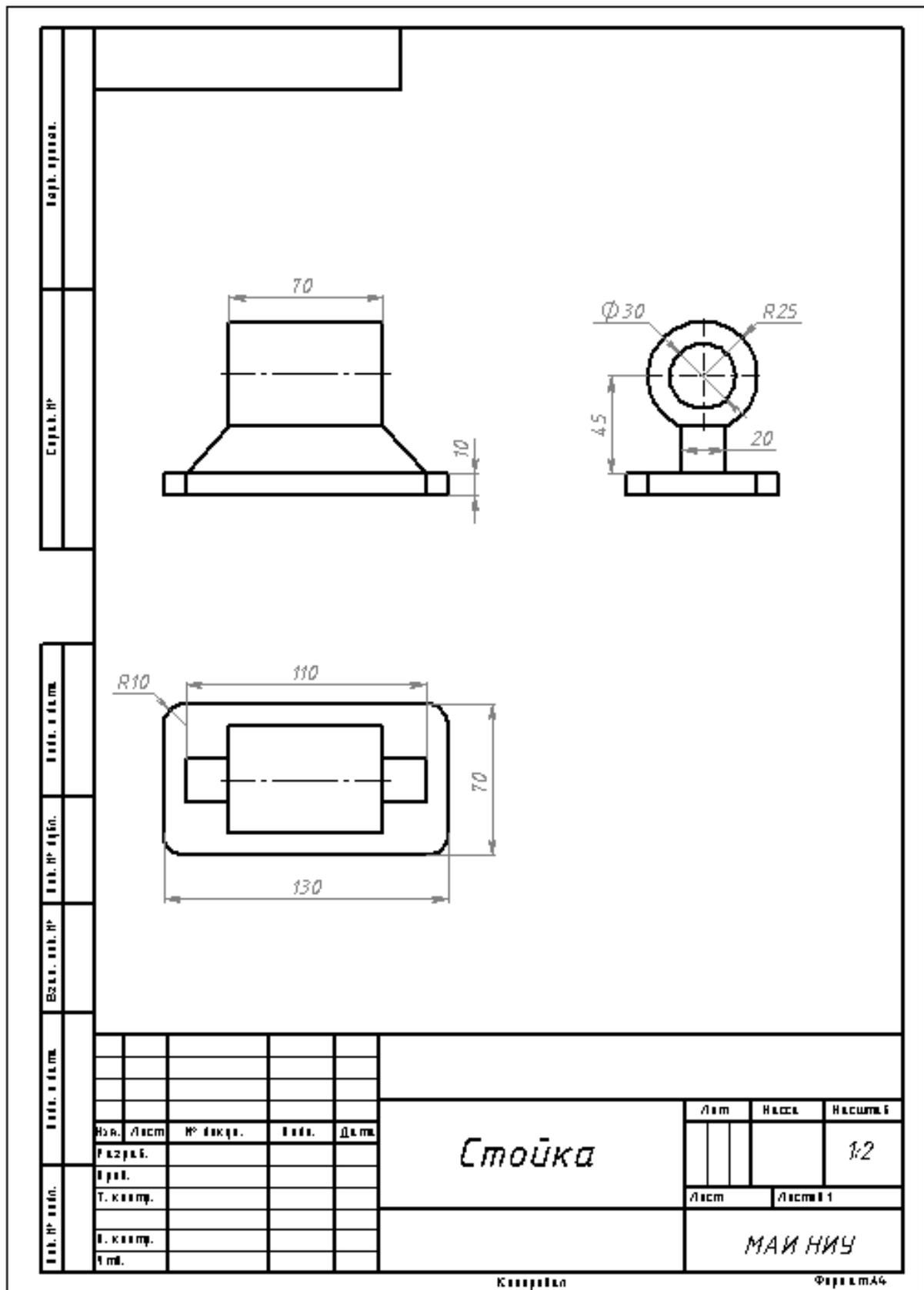


Рисунок 1

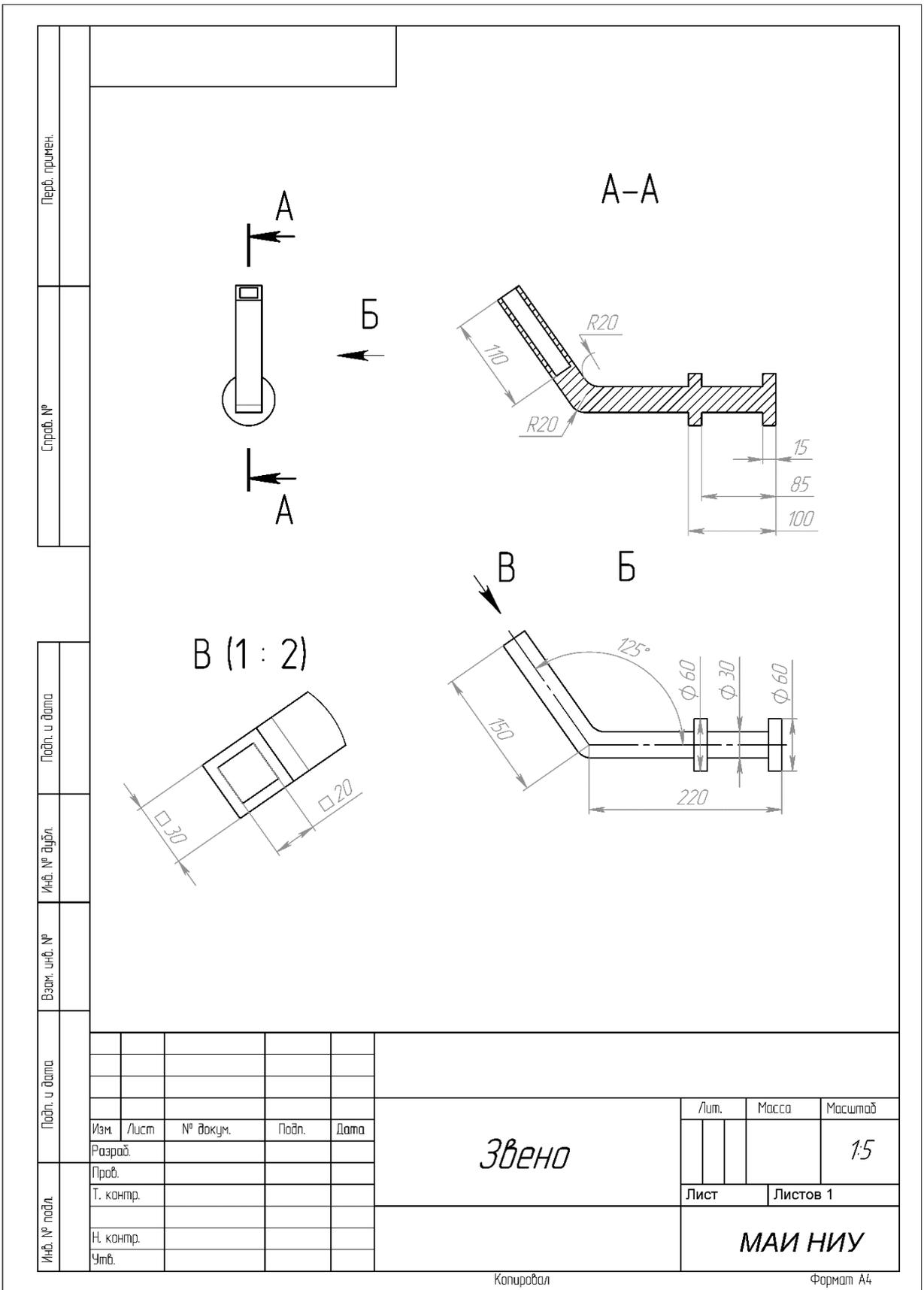


Рисунок 2

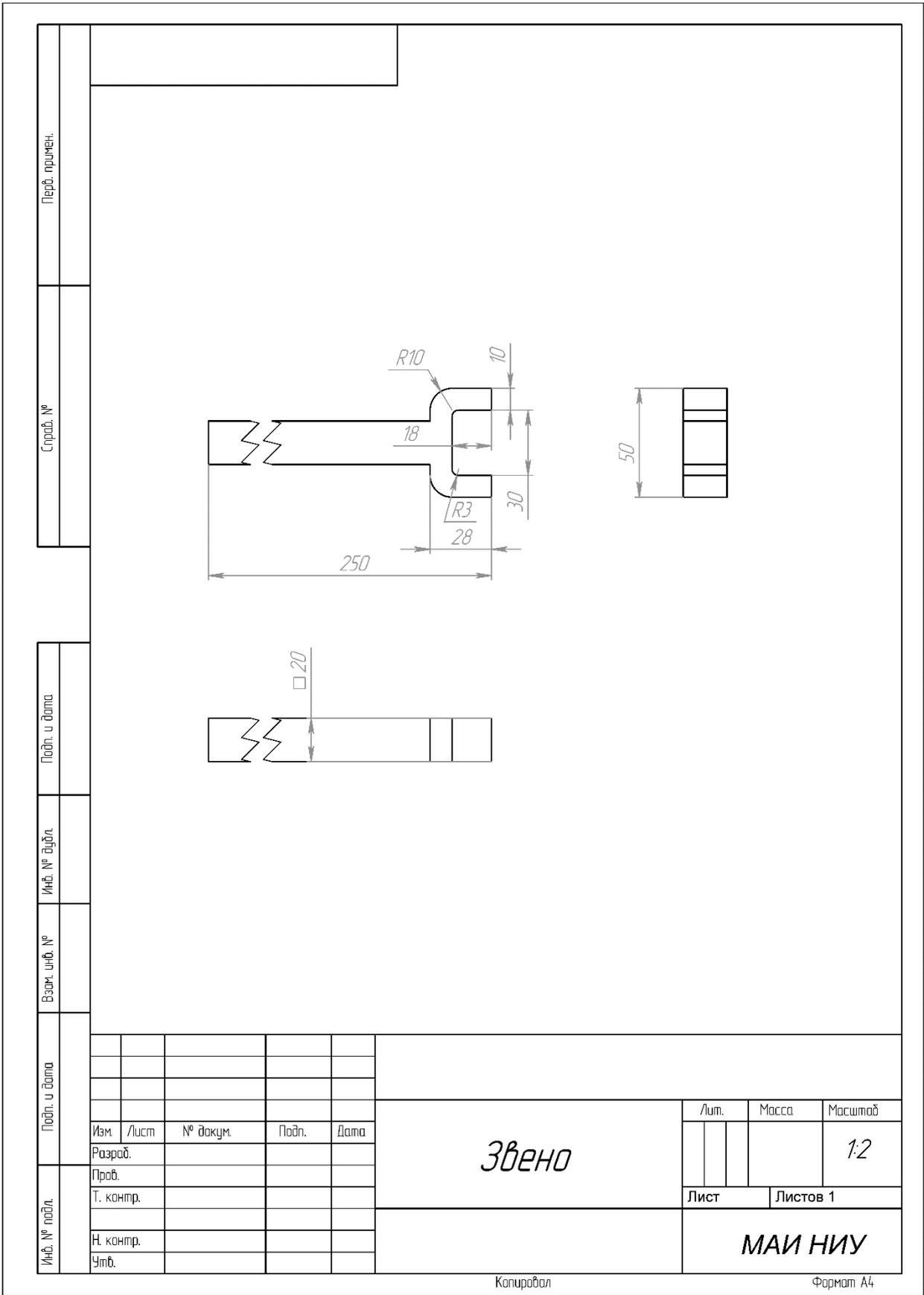


Рисунок 3

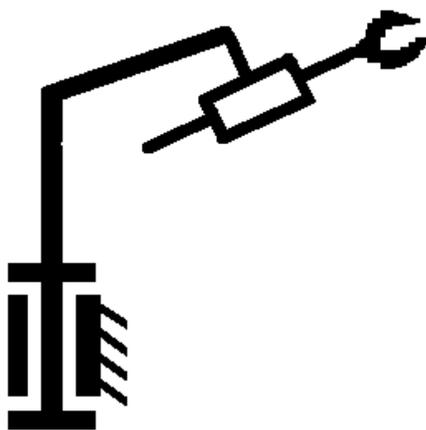


Рисунок 4

## Вариант 9

### Кейс №1.

«Программа учёта космических тел»

Разработать на языке Python программу с консольным интерфейсом для учёта космических тел. Данные о каждом космическом теле представляют собой совокупность следующих значений:

Номер	0	1	2	3	4
Поле	Название	Вид	Предполагаемая масса (мантисса*)	Предполагаемая масса (Степень десятки*)	Текстовая запись

\*Предполагаемая масса складывается из двух частей из <мантиссы>, умноженной на 10 в степени из колонки <степень десятки>

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:

1. **ВПИСАТЬ** <Название> <Вид> <Предполагаемая масса (мантисса)> <Предполагаемая масса (Степень десятки)> <Текстовая запись> – добавляет данные о новом космическом теле с заданными значениями. После выполнения функции в консоль приложения записывается фраза: **ДОБАВЛЕНА НОВАЯ ЗАПИСЬ** <Название>

2. **НАЙТИ** <Название> – выводит в консоль всю информацию о космическом теле с заданным названием. После выполнения функции в консоль приложения записывается фраза: **ИНФОРМАЦИЯ О ЗАПИСИ <Название> ПРЕДСТАВЛЕНА ВЫШЕ**
3. **СТЕРЕТЬ** <Название> – удаляет космическое тело с заданным названием. После выполнения функции в консоль приложения записывается фраза: **ЗАПИСЬ <Название> УДАЛЕНА**
4. **УПОРЯДОЧИТЬ** <номер\_поля\_сорт> – выводит в консоль информацию обо всех добавленных космических телах, 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>**.
5. **ЭКСПОРТ** <имя\_файла> – сохраняет добавленные космические тела в заданный текстовый файл в формате команды **ВПИСАТЬ** (с разделением значений пробелами): <Название> <Вид> <Предполагаемая масса (мантисса)> <Предполагаемая масса (Степень десятки)> <Текстовая запись>. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> **ЗАПИСАН**.
6. **СЧИТАТЬ** <имя\_файла> – загружает данные о космических телах из заданного текстового файла, полностью замещая текущий состав записей о космических телах. Считается, что загружаемый файл был создан при вызове команды **ЭКСПОРТ**. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> **ЗАГРУЖЕН**.

Описание параметров команд:

- <Название> – название космического тела, БЕЗ разделения пробелами (поле №0), например, ТретьяПланета. Данное значение является уникальным для каждого космического тела.
- <Вид> – классификация космического тела, строковое значение, не содержащее пробелов (поле №1).
- <Предполагаемая масса (мантисса)> – характеристика массы космического тела, положительное вещественное число (поле №2).
- <Предполагаемая масса (Степень десятки)> – характеристика массы космического тела, натуральное число (поле №3).
- <Текстовая запись> – текстовая заметка о космическом теле, 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
ВПИСАТЬ Астероид1313 комета 4.3 10 Безопасен	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ Астероид1313
ВПИСАТЬ Kepler5432 планета 7.3 210 ВозможнаЖизнь	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ Kepler5432
ВПИСАТЬ Солнце звезда 2.0 270 БлижайшаяЗвезда	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ Солнце
УПОРЯДОЧИТЬ 3 СТЕРЕТЬ Солнце НАЙТИ Астероид1313 ЭКСПОРТ space.txt СТЕРЕТЬ Kepler5432 УПОРЯДОЧИТЬ 1 СЧИТАТЬ space.txt УПОРЯДОЧИТЬ 0	Астероид1313 комета 4.3 10 Безопасен  Kepler5432 планета 7.3 210 ВозможнаЖизнь  Солнце звезда 0.2 270 БлижайшаяЗвезда  ДАННЫЕ ОТСОРТИРОВАНЫ ПО 3 ЗАПИСЬ Солнце УДАЛЕНА  Астероид1313 комета 4.3 10 Безопасен  ИНФОРМАЦИЯ О ЗАПИСИ Астероид1313 ПРЕДСТАВЛЕНА ВЫШЕ space.txt ЗАПИСАН  ЗАПИСЬ Kepler5432 УДАЛЕНА  Астероид1313 комета 4.3 10 Безопасен  ДАННЫЕ ОТСОРТИРОВАНЫ ПО 1 space.txt ЗАГРУЖЕН  Kepler5432 планета 7.3 210 ВозможнаЖизнь  Астероид1313 комета 4.3 10 Безопасен  ДАННЫЕ ОТСОРТИРОВАНЫ ПО 0

#### 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 не корректно при любом способе реализации – 0 баллов.

№ п/п	Команда	Реализована как функция и	Реализована не как функция и	Выполняется не корректно
----------	---------	------------------------------	---------------------------------	-----------------------------

		<b>выполняется корректно</b>	<b>выполняется корректно</b>	<b>при любом способе реализации</b>
1	ВПИСАТЬ	5 баллов	3 балла	0 баллов
2	НАЙТИ	5 баллов	3 балла	0 баллов
3	СТЕРЕТЬ	5 баллов	3 балла	0 баллов
4	УПОРЯДОЧИТ Ь	5 баллов	3 балла	0 баллов
5	ЭКСПОРТ	5 баллов	3 балла	0 баллов
6	СЧИТАТЬ	5 баллов	3 балла	0 баллов
<b>ИТОГО</b>		<b>30 баллов</b>	<b>18 баллов</b>	<b>0 баллов</b>

## Кейс №2.

### *Задание 1*

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

### *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.

### *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.

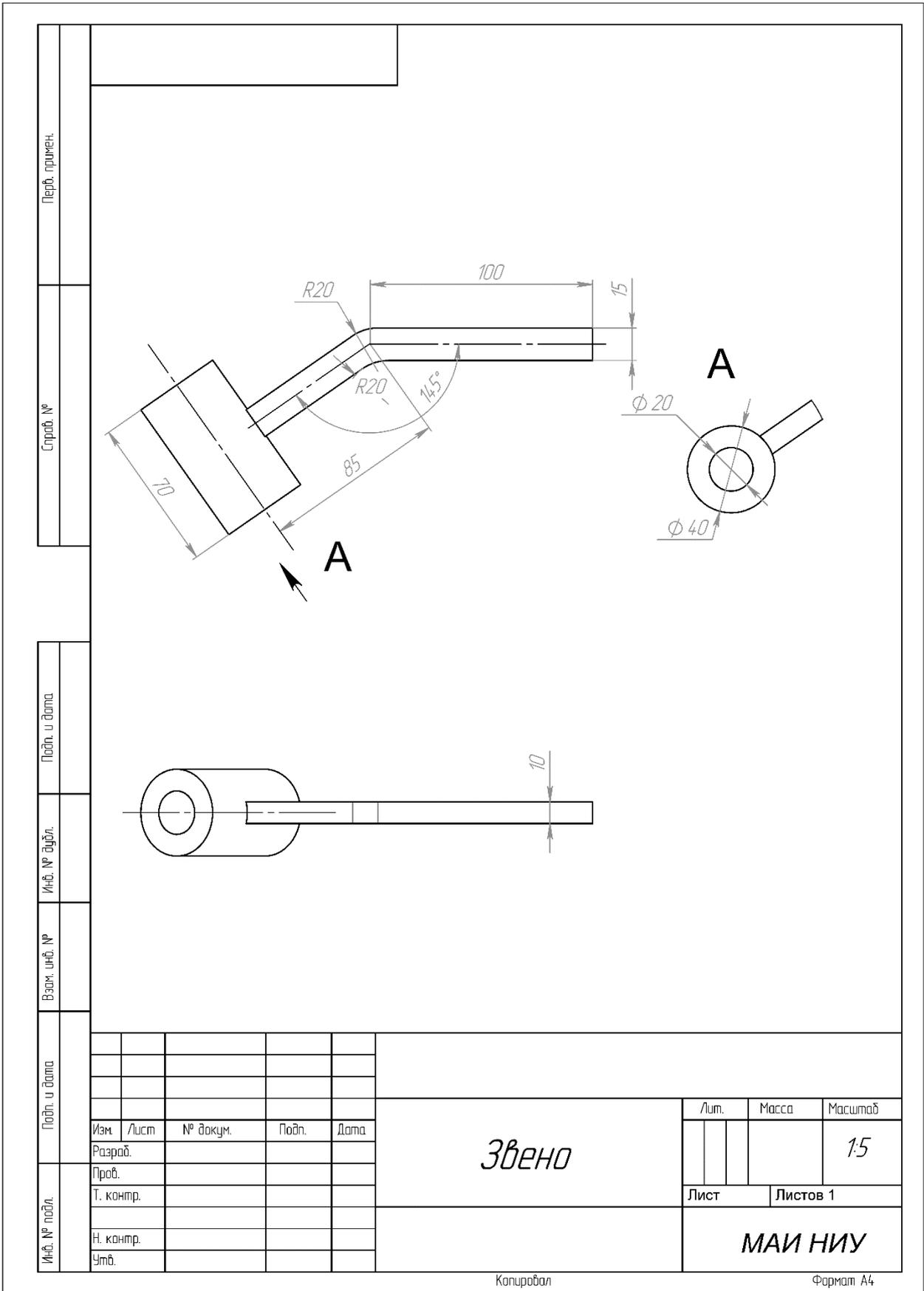


Рисунок 1

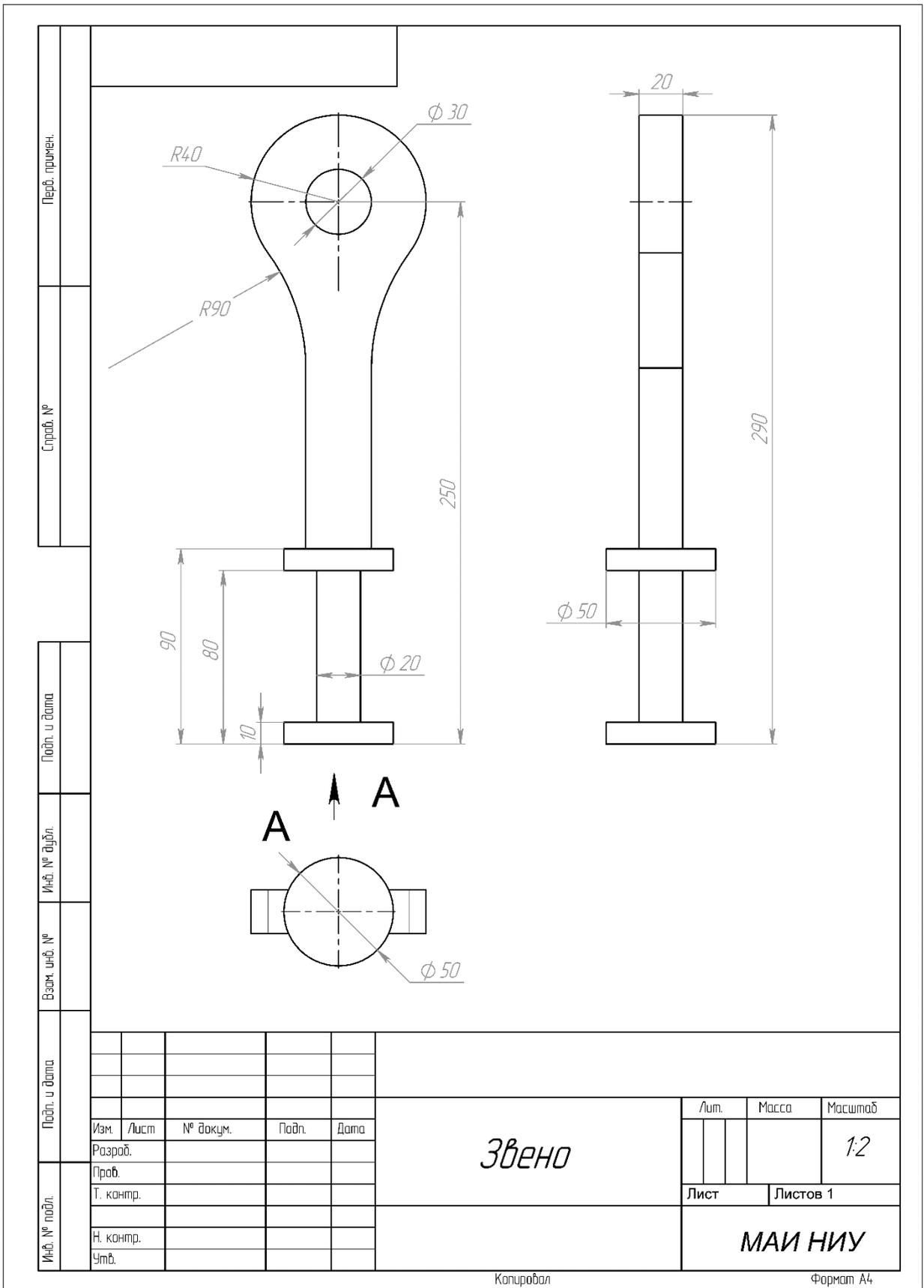
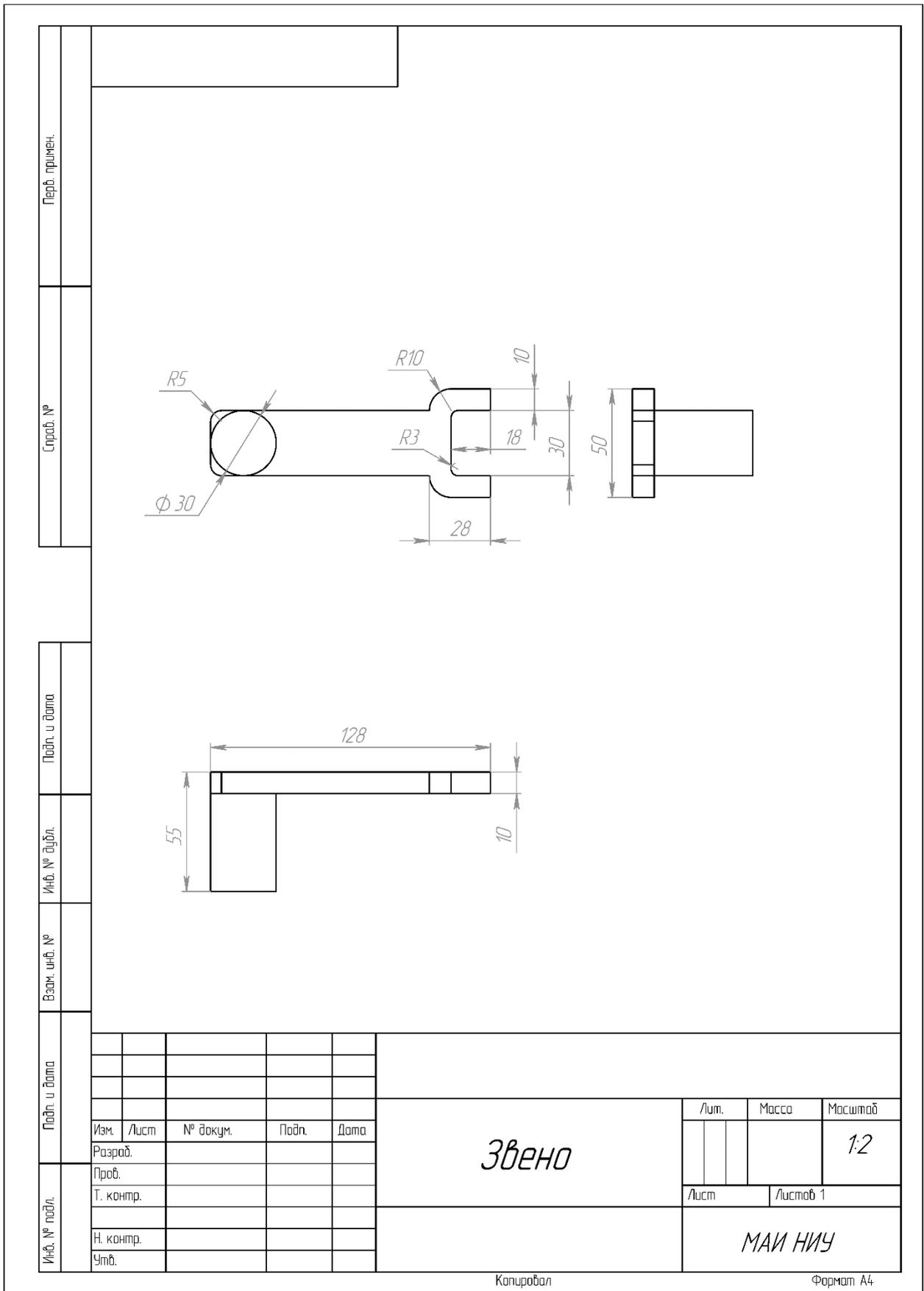


Рисунок 2



Копировал

Формат А4

Рисунок 3

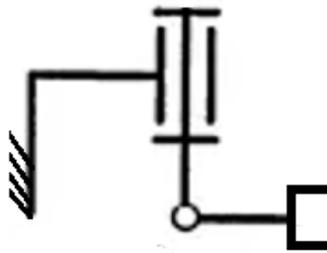


Рисунок 4

## Вариант 10

### Кейс №1.

«Программа учёта недвижимости»

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

Номер	0	1	2	3	4
Поле	Название	Владелец	Год приобретения	Тип недвижимости	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:

1. **ВВЕСТИ** <Название> <Владелец> <Год приобретения> <Тип недвижимости> <Текстовая заметка> – добавляет данные о новой недвижимости с заданными значениями. После выполнения функции в консоль приложения записывается фраза: **ДОБАВЛЕНА НОВАЯ ЗАПИСЬ** <Название>
2. **ВЫВЕСТИ** <Название> – выводит в консоль всю информацию о недвижимости с заданным названием. После выполнения функции в консоль приложения записывается фраза: **ИНФОРМАЦИЯ О ЗАПИСИ** <Название> **ПРЕДСТАВЛЕНА ВЫШЕ**
3. **СТЕРЕТЬ** <Название> – удаляет недвижимость с заданным названием. После выполнения функции в консоль приложения записывается фраза: **ЗАПИСЬ** <Название> **УДАЛЕНА**

4. СОРТИРОВАТЬ <номер\_поля\_сорт> – выводит в консоль информацию обо всей добавленной недвижимости, 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>.
5. ЗАПИСАТЬ <имя\_файла> – сохраняет добавленную недвижимость в заданный текстовый файл в формате команды ВВЕСТИ (с разделением значений пробелами) : <Название> <Владелец> <Год приобретения> <Тип недвижимости> <Текстовая заметка>. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> ЗАПИСАН.
6. СЧИТАТЬ <имя\_файла> – загружает данные о недвижимости 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 недвижимости БЕЗ разделения пробелами (поле №0), например, Квартира255. Данное значение является уникальным для каждой недвижимости.
- <Владелец> – фамилия имя владельца недвижимости, строковое значение, не содержащее пробелов (поле №1).
- <Год приобретения> – год приобретения недвижимости, натуральное четырёхзначное число (поле №2).
- <Тип недвижимости> – строковое значение, не содержащее пробелов (поле №3).
- <Текстовая заметка> – текстовая заметка о недвижимости, 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 Квартира255 ИвановИ.И 1990 Жилая БольшаяКвартира	ДОБАВЛЕНА НОВЫЙ ЗАПИСЬ Квартира255
ВВЕСТИ Помещение1 ПетровП.А 2020 Коммерческая ПодТорговлю	ДОБАВЛЕНА НОВЫЙ ЗАПИСЬ Помещение1
ВВЕСТИ Дом256 СидоровА.А 2005 Жилая ДомСУчастком	ДОБАВЛЕНА НОВЫЙ ЗАПИСЬ Дом256

СОРТИРОВАТЬ 2	Квартира255 ИвановИ.И 1990 Жилая
УБРАТЬ Квартира255	БольшаяКвартира
ВЫВЕСТИ Помещение1	Дом256 СидороваА.А 2005 Жилая
ЗАПИСАТЬ estates.txt	ДомСУчастком
УБРАТЬ Помещение1	Помещение1 ПетровП.А 2020
СОРТИРОВАТЬ 1	Коммерческая ПодТорговлю
СЧИТАТЬ estates.txt	ДААННЫЕ ОТСОРТИРОВАНЫ ПО 2
СОРТИРОВАТЬ 0	ЗАПИСЬ Квартира255 УДАЛЕНА
	Помещение1 ПетровП.А 2020
	Коммерческая ПодТорговлю
	ИНФОРМАЦИЯ О ЗАПИСИ
	Помещение1 ПРЕДСТАВЛЕНА ВЫШЕ
	estates.txt ЗАПИСАН
	ЗАПИСЬ Помещение1 УДАЛЕНА
	Дом256 СидороваА.А 2005 Жилая
	ДомСУчастком
	ДААННЫЕ ОТСОРТИРОВАНЫ ПО 1
	estates.txt ЗАГРУЖЕН
	Дом256 СидороваА.А 2005 Жилая
	ДомСУчастком
	Квартира255 ИвановИ.И 1990 Жилая
	БольшаяКвартира
	ДААННЫЕ ОТСОРТИРОВАНЫ ПО 0

### 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	Реализована не как функция и	Выполняется не корректно
-------	---------	-------------------------	------------------------------	--------------------------

		<b>и выполняется корректно</b>	<b>выполняется корректно</b>	<b>при любом способе реализации</b>
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
<b>ИТОГО</b>		<b>30 баллов</b>	<b>18 баллов</b>	<b>0 баллов</b>

## Кейс №2.

### *Задание 1*

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

### *Задание 2*

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

### *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. Экпортируйте сборку в формат .stl. – 5 баллов.

Перв. примен.																																																															
Спроб. №																																																															
Подп. и дата																																																															
Инв. № дубл.																																																															
Взам. инв. №	<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 10%;"></td> </tr> <tr> <td>Изм.</td> <td>Лист</td> <td>№ докум.</td> <td>Подп.</td> <td>Дата</td> <td colspan="5" rowspan="2" style="text-align: center; vertical-align: middle;"> <h1 style="font-size: 2em;">Основание</h1> </td> <td style="width: 10%; text-align: center;">Лит.</td> <td style="width: 10%; text-align: center;">Масса</td> <td style="width: 10%; text-align: center;">Масштаб</td> </tr> <tr> <td>Разраб.</td> <td></td> <td></td> <td></td> <td></td> <td style="text-align: center;">1:2</td> </tr> <tr> <td>И/в. № подл.</td> <td>Проб.</td> <td colspan="8" rowspan="3" style="text-align: center; vertical-align: middle;"> <h1 style="font-size: 2em;">МАИ НИУ</h1> </td> <td colspan="2" style="text-align: center;">Лист</td> <td style="text-align: center;">Листов 1</td> </tr> <tr> <td></td> <td>Т. контр.</td> </tr> <tr> <td></td> <td>Н. контр.</td> </tr> <tr> <td></td> <td>Утв.</td> <td></td> </tr> </table>														Изм.	Лист	№ докум.	Подп.	Дата	<h1 style="font-size: 2em;">Основание</h1>					Лит.	Масса	Масштаб	Разраб.					1:2	И/в. № подл.	Проб.	<h1 style="font-size: 2em;">МАИ НИУ</h1>								Лист		Листов 1		Т. контр.		Н. контр.		Утв.											
Изм.	Лист	№ докум.	Подп.	Дата	<h1 style="font-size: 2em;">Основание</h1>					Лит.	Масса	Масштаб																																																			
Разраб.										1:2																																																					
И/в. № подл.	Проб.	<h1 style="font-size: 2em;">МАИ НИУ</h1>								Лист		Листов 1																																																			
	Т. контр.																																																														
	Н. контр.																																																														
	Утв.																																																														
Подп. и дата	<p style="text-align: center;">Копировал</p> <p style="text-align: right;">Формат А4</p>																																																														
Инв. № подл.																																																															

Рисунок 1

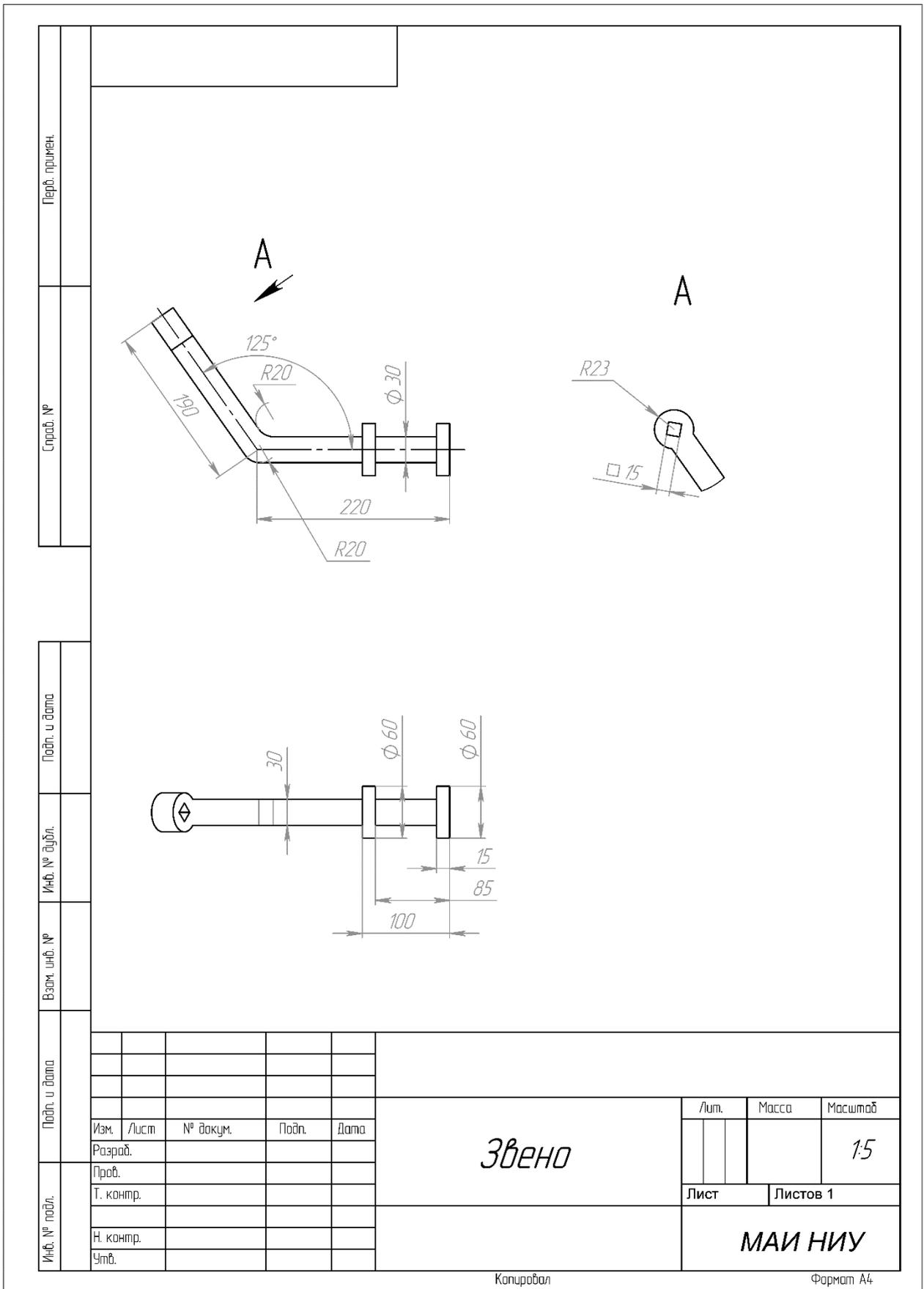


Рисунок 2

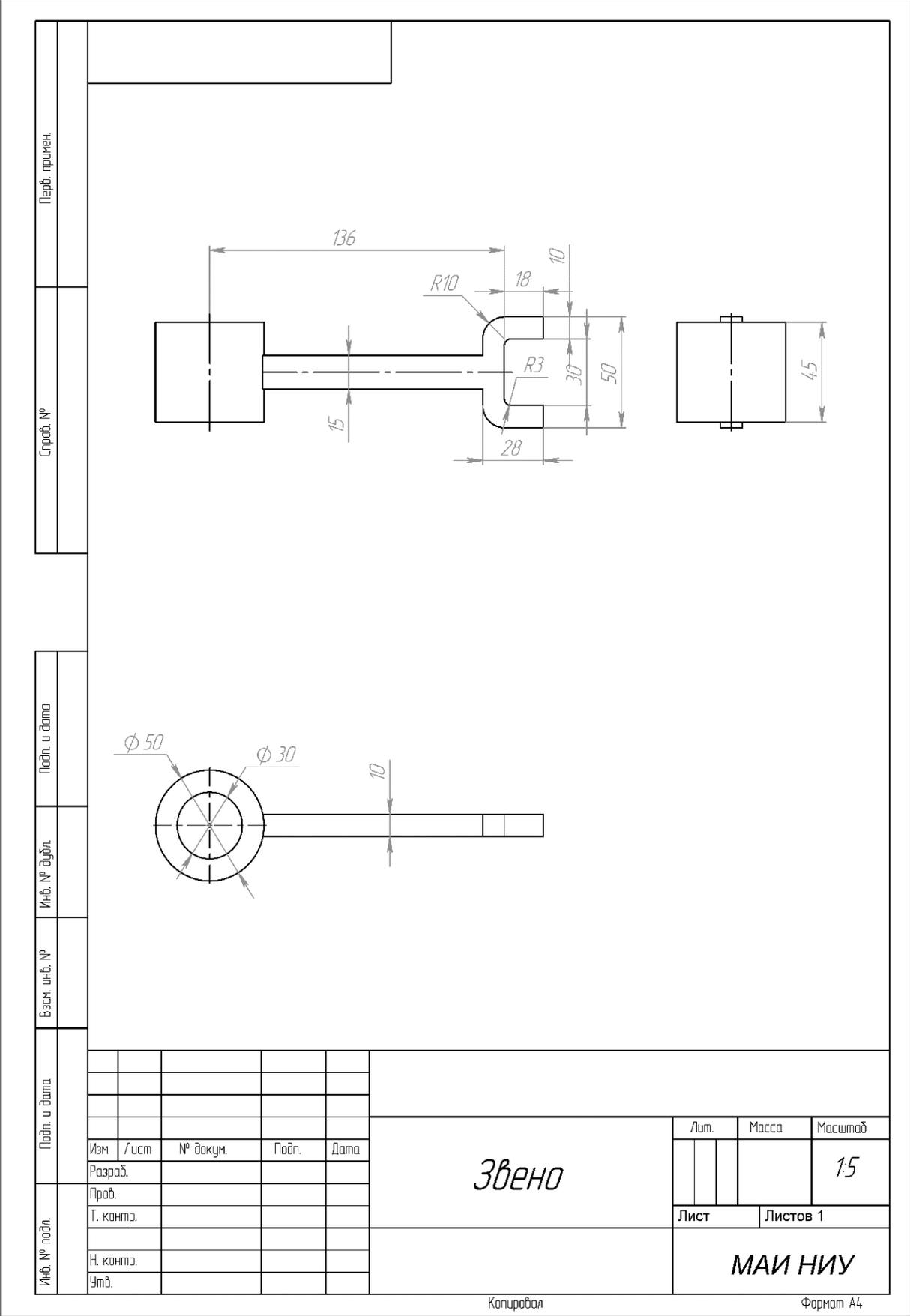


Рисунок 3

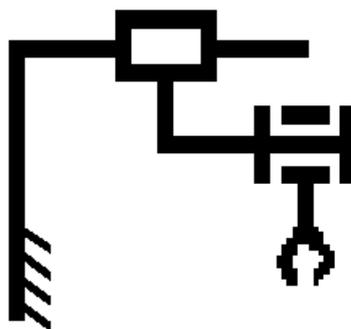


Рисунок 4

## Вариант 11

### Кейс №1.

«Программа учёта химических реагентов на складах»

Разработать на языке Python программу с консольным интерфейсом для учёта химических реагентов в лаборатории. Данные о каждом реагенте представляют собой совокупность следующих значений:

Номер	0	1	2	3	4
Поле	Название	Молярная масса	Запас в граммах	Вид	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:

1. **ВПИСАТЬ** <Название> <Молярная масса> <Запас в граммах> <Вид> <Текстовая заметка> – добавляет данные о новом реагенте с заданными значениями. После выполнения функции в консоль приложения записывается фраза: **ДОБАВЛЕНА НОВАЯ ЗАПИСЬ** <Название>
2. **ОТОБРАЗИТЬ** <Название> – выводит в консоль всю информацию о реагентах с заданным названием. После выполнения функции в консоль приложения записывается фраза: **ИНФОРМАЦИЯ О ЗАПИСИ** <Название> **ПРЕДСТАВЛЕНА ВЫШЕ**
3. **УБРАТЬ** <Название> – удаляет реактив с заданным названием. После выполнения функции в консоль приложения записывается фраза: **ЗАПИСЬ** <Название> **УДАЛЕН**

4. УПОРЯДОЧИТЬ <номер\_поля\_сорт> – выводит в консоль информацию обо всех добавленных реагентах, 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>.
5. СОЗДАТЬ\_ФАЙЛ <имя\_файла> – сохраняет добавленные реагенты в заданный текстовый файл в формате команды ВПИСАТЬ (с разделением значений пробелами): <Название> <Молярная масса> <Запас в граммах> <Вид> <Текстовая заметка>. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> ЗАПИСАН.
6. ПРОЧИТАТЬ\_ФАЙЛ <имя\_файла> – загружает данные о реагентах из заданного текстового файла, полностью замещая текущий состав записей о реагентах. Считается, что загружаемый файл был создан при вызове команды СОЗДАТЬ\_ФАЙЛ. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> ЗАГРУЖЕН.

Описание параметров команд:

- <Название> – Название реагента, БЕЗ разделения пробелами (поле №0), например, ХлоридНатрия. Данное значение является уникальным для каждого реагента.
- <Молярная масса> – характеристика реагента, вещественное значение, (поле №1).
- <Запас в граммах> – количество реагента, натуральное число (поле №2).
- <Вид> – классификация реагента, строковое значение, не содержащее пробелов (поле №3).
- <Текстовая заметка> – текстовая заметка о реагенте, 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
ВПИСАТЬ БромидКалия 119.002 115 Неорганическая БелыйПорошок	ДОБАВЛЕНА НОВЫЙ ЗАПИСЬ БромидКалия
ВПИСАТЬ ХлоридЖелеза3 58.44 25 Неорганическая БелыйКристалл	ДОБАВЛЕНА НОВЫЙ ЗАПИСЬ ХлоридЖелеза3
ВПИСАТЬ Глюкоза 180.156 200 Органическая Сладкая	ДОБАВЛЕНА НОВЫЙ ЗАПИСЬ Глюкоза

УПОРЯДОЧИТЬ 2 УБРАТЬ Глюкоза ОТОБРАЗИТЬ БромидКалия СОЗДАТЬ_ФАЙЛ reactives.txt УБРАТЬ ХлоридЖелеза3 УПОРЯДОЧИТЬ 1 ПРОЧИТАТЬ_ФАЙЛ reactives.txt УПОРЯДОЧИТЬ 0	ХлоридЖелеза3 58.44 25 Неорганическая БелыйКристалл БромидКалия 119.002 115 Неорганическая БелыйПорошок Глюкоза 180.156 200 Органическая Сладкая ЗАПИСЬ Глюкоза УДАЛЕН БромидКалия 119.002 115 Неорганическая БелыйПорошок ИНФОРМАЦИЯ О ЗАПИСИ БромидКалия ПРЕДСТАВЛЕНА ВЫШЕ reactives.txt ЗАПИСАН ЗАПИСЬ ХлоридЖелеза3 УДАЛЕН БромидКалия 119.002 115 Неорганическая БелыйПорошок ДАННЫЕ ОТСОРТИРОВАНЫ ПО 1 reactives.txt ЗАГРУЖЕН БромидКалия 119.002 115 Неорганическая БелыйПорошок ХлоридЖелеза3 58.44 25 Неорганическая БелыйКристалл ДАННЫЕ ОТСОРТИРОВАНЫ ПО 0
---	---

### 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	ВПИСАТЬ	5 баллов	3 балла	0 баллов
2	ОТОБРАЗИТЬ	5 баллов	3 балла	0 баллов
3	УБРАТЬ	5 баллов	3 балла	0 баллов
4	УПОРЯДОЧИТЬ	5 баллов	3 балла	0 баллов
5	СОЗДАТЬ_ФАЙЛ	5 баллов	3 балла	0 баллов
6	ПРОЧИТАТЬ_ФАЙЛ	5 баллов	3 балла	0 баллов
<b>ИТОГО</b>		<b>30 баллов</b>	<b>18 баллов</b>	<b>0 баллов</b>

## Кейс №2.

### *Задание 1*

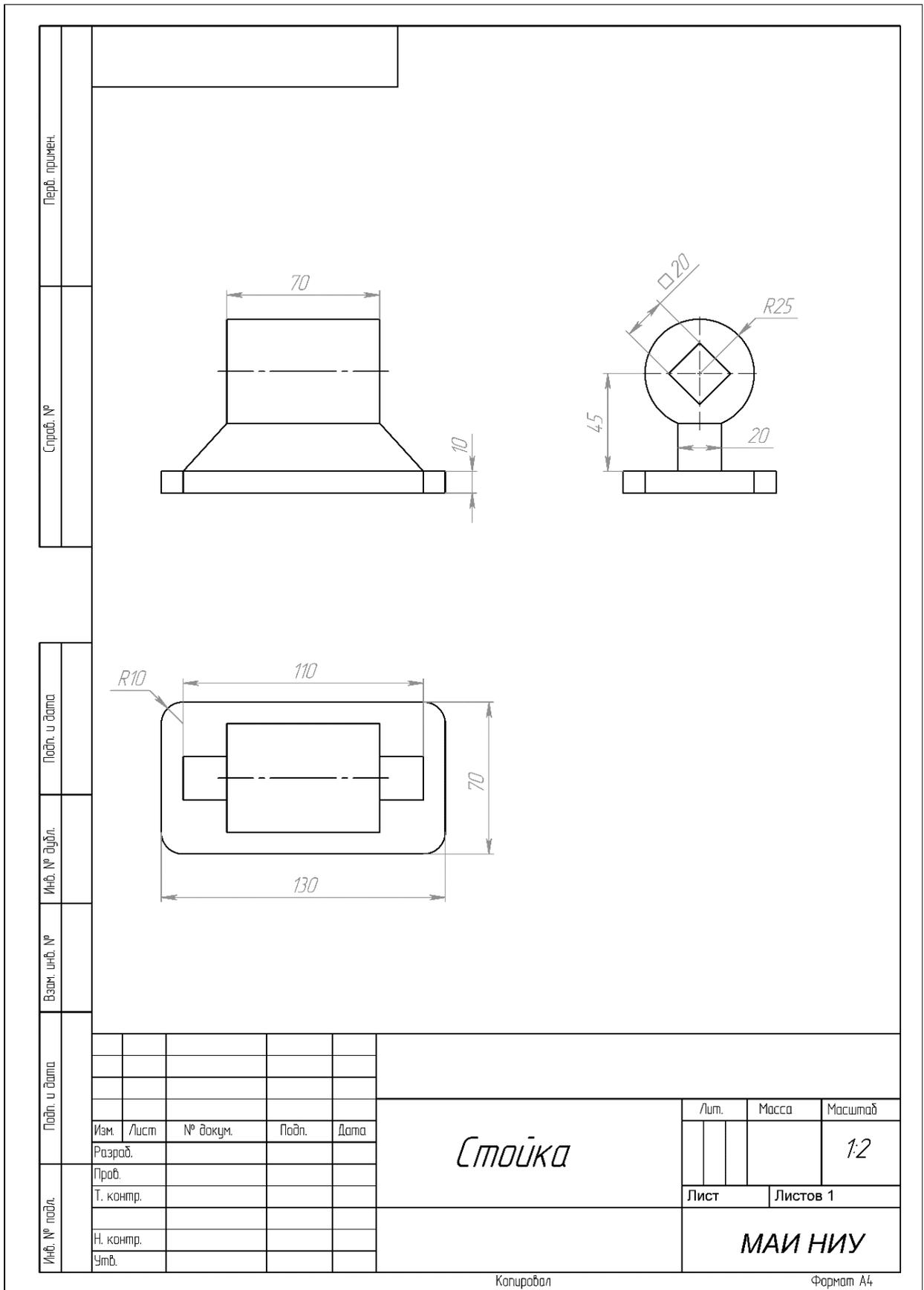
Постройте 3D модели звеньев робота согласно чертежам, представленным на рисунках 1 – 3. Экпортируйте детали в формат .stl. – 15 баллов.

### *Задание 2*

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

### *Задание 3*

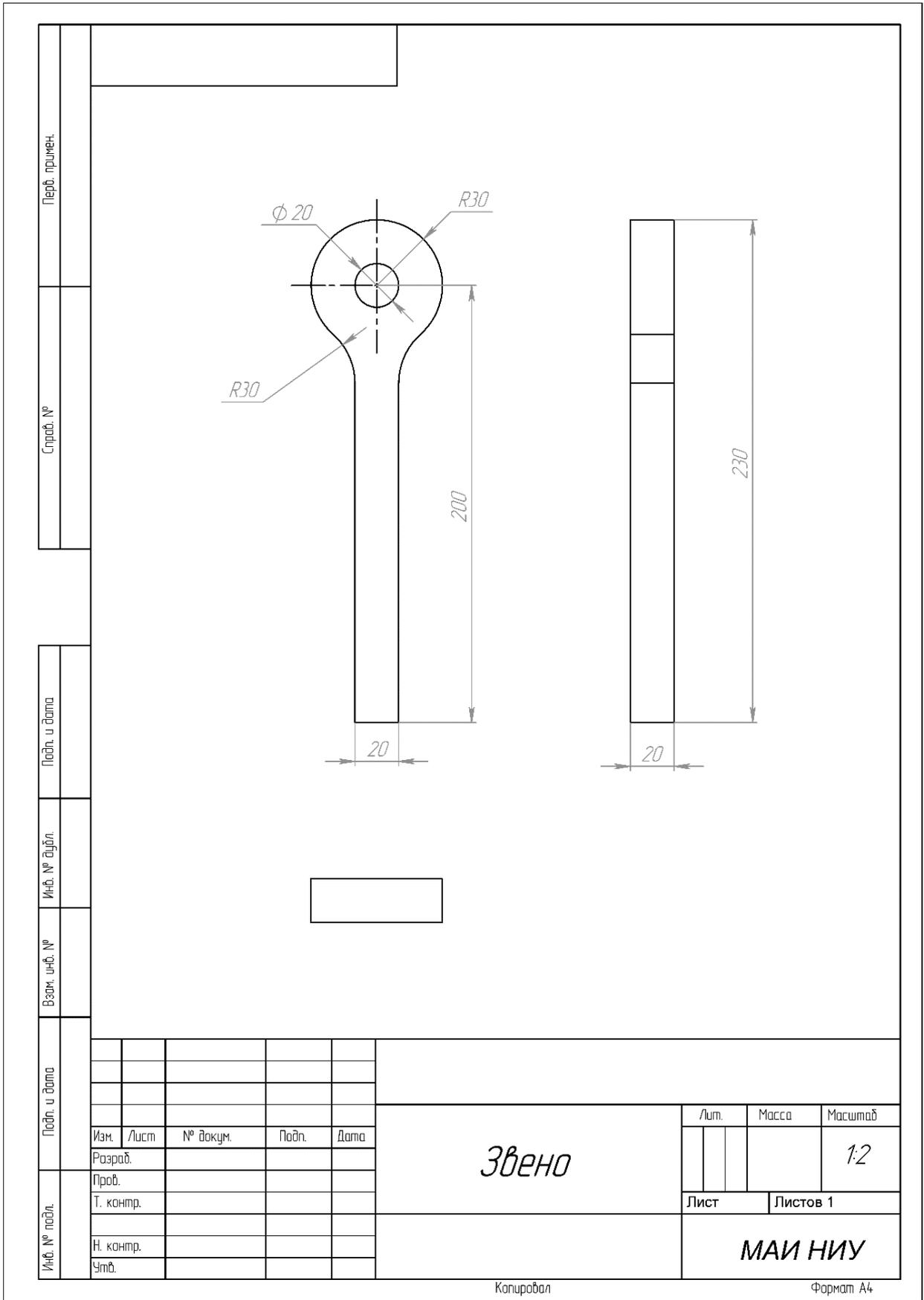
Создайте сборку, имитирующую рациональное расположение и ориентацию деталей при печати на 3D принтере на его рабочем столе. За поверхность рабочего стола принтера принимается базовая горизонтальная плоскость (Сверху. Тор, Вид сверху – в зависимости от выбранной САПР) Критерием рациональности выступает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



Копировал

Формат А4

Рисунок 1



Перв. примен.	
Справ. №	

Подп. и дата	
Инв. № эъл.	
Взам. инв. №	

Подп. и дата					
Изм.	Лист	№ докум.	Подп.	Дата	
Разраб.					
Проб.					
Т. контр.					
Н. контр.					
Утв.					

*Звено*

Лит.	Масса	Масштаб
		1:2
Лист	Листов 1	
<b>МАИ НИУ</b>		

Копировал

Формат А4

Рисунок 2



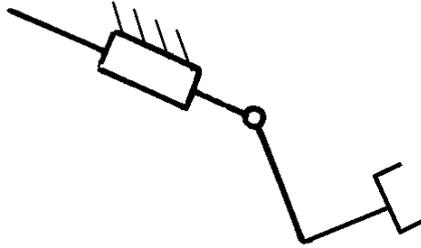


Рисунок 4

## Вариант 12

### Кейс №1.

«Программа учёта исторических событий стран»

Разработать на языке Python программу с консольным интерфейсом для хранения записей о ключевых исторических событиях. Данные о каждом событии представляют собой совокупность следующих значений:

Номер	0	1	2	3	4
Поле	Название	Страна	Часть света	Дата	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:

1. **ЗАДАТЬ** <Название> <Страна> <Часть света> <Дата> <Текстовая заметка> – добавляет данные о новом историческом событии с заданными значениями. После выполнения функции в консоль приложения записывается фраза: **ДОБАВЛЕНА НОВАЯ ЗАПИСЬ** <Название>
2. **НАЙТИ** <Название> – выводит в консоль всю информацию о событии с заданным названием. После выполнения функции в консоль приложения записывается фраза: **ИНФОРМАЦИЯ О ЗАПИСИ** <Название> **ПРЕДСТАВЛЕНА ВЫШЕ**
3. **ИСКЛЮЧИТЬ** <Название> – удаляет событие с заданным названием. После выполнения функции в консоль приложения записывается фраза: **ЗАПИСЬ** <Название> **УДАЛЕНА**
4. **СПИСОК** <номер\_поля\_сорт> – выводит в консоль информацию обо всех добавленных событиях, 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>.

5. ЭКСПОРТ <имя\_файла> – сохраняет добавленные события в заданный текстовый файл в формате команды ЗАДАТЬ (с разделением значений пробелами): <Название> <Страна> <Часть света> <Дата> <Текстовая заметка>. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> ЗАПИСАН
6. ИМПОРТ <имя\_файла> – загружает данные о событиях из заданного текстового файла, полностью замещая текущий состав записей об исторических событиях. Считается, что загружаемый файл был создан при вызове команды ЭКСПОРТ. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> ЗАГРУЖЕН.

Описание параметров команд:

- <Название> – название события, БЕЗ разделения пробелами (поле №0), например, КрещениеРуси. Данное значение является уникальным для каждого события.
- <Страна> – страна, к которой относится событие, строковое значение (поле №1).
- <Часть света> – место на планете, где произошло событие, строковое значение, не содержащее пробелов (поле №2).
- <Дата> – год, когда произошло событие, натуральное четырёхзначное число (поле №3).
- <Текстовая заметка> – текстовая заметка о событии, 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
ЗАДАТЬ ПокорениеСибири Россия Азия 1582 Быстроепокорение	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ ПокорениеСибири
ЗАДАТЬ ПадениеПольши Польша Европа 1795 Внезапно	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ ПадениеПольши
ЗАДАТЬ ОсвобождениеКрестьян Россия Евразия 1861 Свободныелюди	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ ОсвобождениеКрестьян
СПИСОК 3 ИСКЛЮЧИТЬ ПадениеПольши НАЙТИ ПокорениеСибири	ПокорениеСибири Россия Азия 1582 Быстроепокорение

<p>ЭКСПОРТ history.txt  ИСКЛЮЧИТЬ ОсвобождениеКрестьян  СПИСОК 1  ИМПОРТ history.txt  СПИСОК 0</p>	<p>ПадениеПольши Польша Европа 1795  Внезапно  ОсвобождениеКрестьян Россия  Евразия 1861 Свободныелюди  ДАнные ОТСОРТИРОВАНЫ ПО 3  ЗАПИСЬ ПадениеПольши УДАЛЕНА  ПокорениеСибири Россия Азия 1582  Быстроепокорение  ИНФОРМАЦИЯ О ЗАПИСИ  ПокорениеСибири ПРЕДСТАВЛЕНА  ВЫШЕ  history.txt ЗАПИСАН  ЗАПИСЬ ОсвобождениеКрестьян  УДАЛЕНА  ПокорениеСибири Россия Азия 1582  Быстроепокорение  ДАнные ОТСОРТИРОВАНЫ ПО 1  history.txt ЗАГРУЖЕН  ОсвобождениеКрестьян Россия  Евразия 1861 Свободныелюди  ПокорениеСибири Россия Азия 1582  Быстроепокорение  ДАнные ОТСОРТИРОВАНЫ ПО 0</p>
--	--

### 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 не корректно 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	ЗАДАТЬ	5 баллов	3 балла	0 баллов

2	НАЙТИ	5 баллов	3 балла	0 баллов
3	ИСКЛЮЧИТЬ	5 баллов	3 балла	0 баллов
4	СПИСОК	5 баллов	3 балла	0 баллов
5	ЭКСПОРТ	5 баллов	3 балла	0 баллов
6	ИМПОРТ	5 баллов	3 балла	0 баллов
<b>ИТОГО</b>		<b>30 баллов</b>	<b>18 баллов</b>	<b>0 баллов</b>

## Кейс №2.

### *Задание 1*

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

### *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.

### *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.

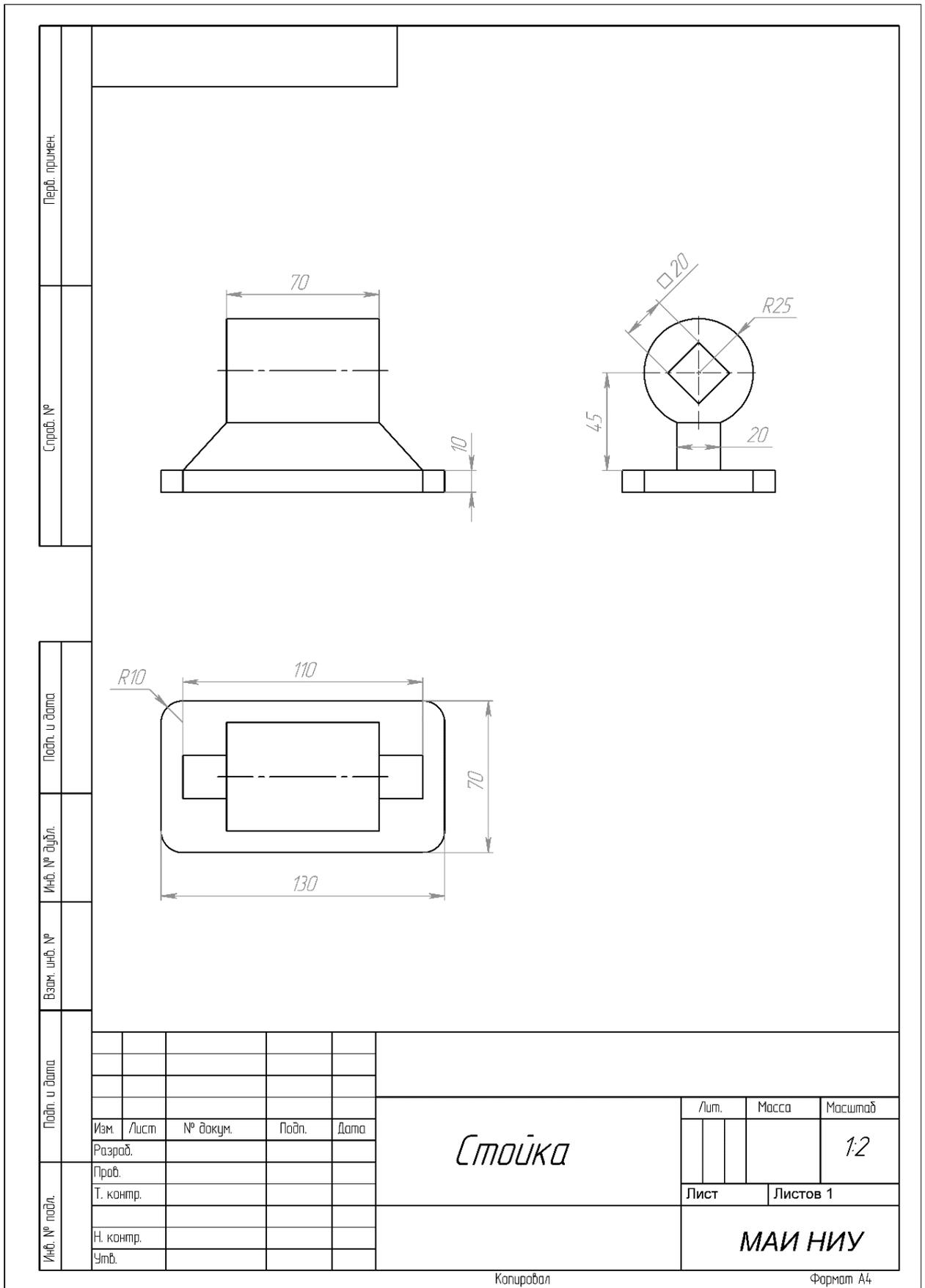
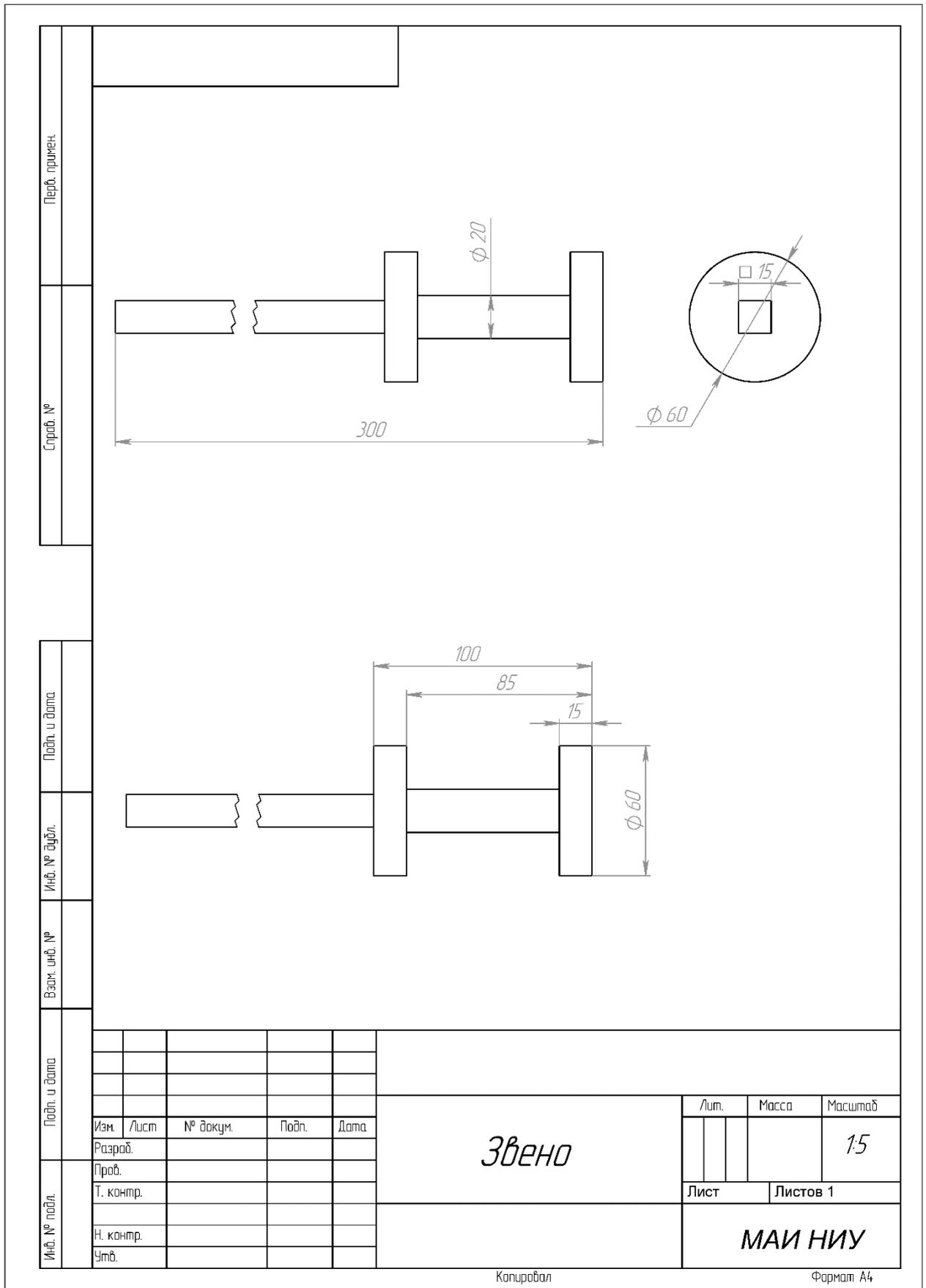


Рисунок 1



Перв. примен.  
Спроб. №

Подп. и дата  
Инв. № табл.  
Взам. инв. №

Подп. и дата  
Инв. № табл.

Изм.	Лист	№ докум.	Подп.	Дата
Разраб.				
Проб.				
Т. контр.				
Н. контр.				
Утв.				

*Звено*

Лит.	Масса	Масштаб
		1:5
Лист		Листов 1
<b>МАИ НИУ</b>		

Копировал

Формат А4

Рисунок 2

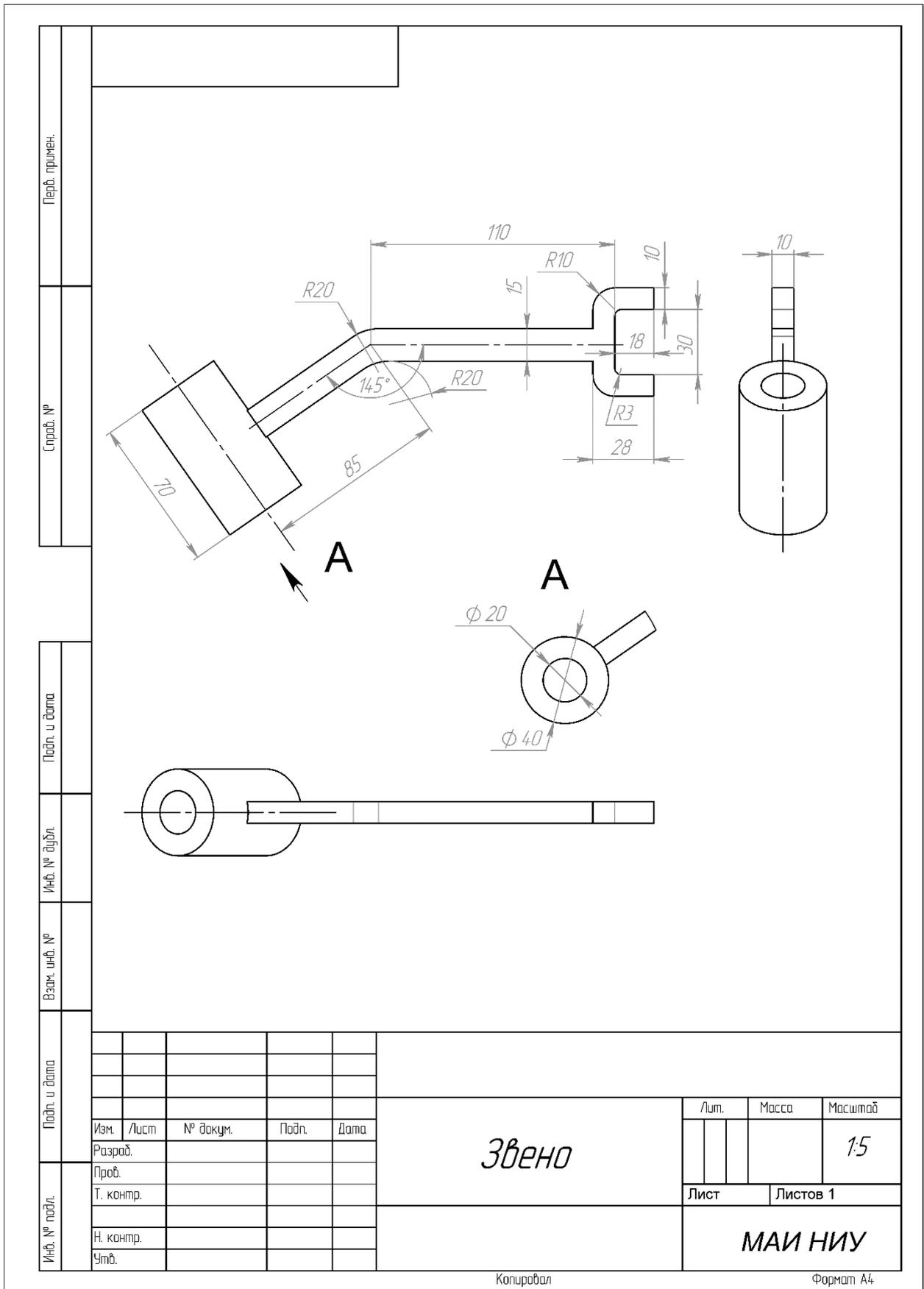


Рисунок 3

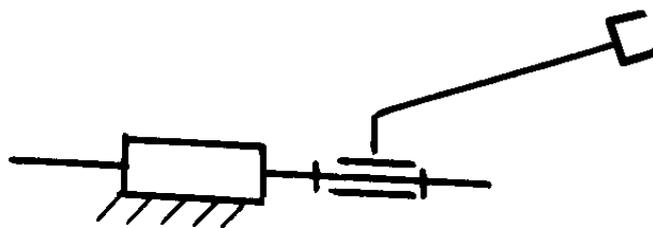


Рисунок 4

## Вариант 13

### Кейс №1.

«Программа учёта ингредиентов в ресторане»

Разработать на языке Python программу с консольным интерфейсом для учёта ингредиентов. Данные о каждом ингредиенте представляют собой совокупность следующих значений:

Номер	0	1	2	3	4
Поле	Название	Цвет	Вкус	Запас в килограммах	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:

1. **ВВЕСТИ** <Название> <Цвет> <Вкус> <Запас в килограммах> <Текстовая заметка> – добавляет данные о новом ингредиенте с заданными значениями. После выполнения функции в консоль приложения записывается фраза: **ДОБАВЛЕНА НОВАЯ ЗАПИСЬ** <Название>
2. **ОТОБРАЗИТЬ** <Название> – выводит в консоль всю информацию об ингредиенте с заданным названием. После выполнения функции в консоль приложения записывается фраза: **ИНФОРМАЦИЯ О ЗАПИСИ** <Название> **ПРЕДСТАВЛЕНА ВЫШЕ**
3. **ИСКЛЮЧИТЬ** <Название> – удаляет ингредиент с заданным названием. После выполнения функции в консоль приложения записывается фраза: **ЗАПИСЬ** <Название> **УДАЛЕНА**
4. **СОРТИРОВАТЬ** <номер\_поля\_сорт> – выводит в консоль информацию обо всех добавленных ингредиентах, 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>.

5. СОЗДАТЬ\_ФАЙЛ <имя\_файла> – сохраняет добавленные ингредиенты в заданный текстовый файл в формате команды ВВЕСТИ (с разделением значений пробелами): <Название> <Цвет> <Вкус> <Запас в килограммах> <Текстовая заметка>. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> ЗАПИСАН.
6. ИМПОРТ <имя\_файла> – загружает данные об ингредиентах из заданного текстового файла, полностью замещая текущий состав записей об ингредиентах. Считается, что загружаемый файл был создан при вызове команды СОЗДАТЬ\_ФАЙЛ. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> ЗАГРУЖЕН.

Описание параметров команд:

- <Название> – Название ингредиента, БЕЗ разделения пробелами (поле №0), например, ЧернаяСмородина. Данное значение является уникальным для каждого ингредиента.
- <Цвет> – характеристика ингредиента, строковое значение, не содержащее пробелов (поле №1).
- <Вкус> – характеристика ингредиента, строковое значение, не содержащее пробелов (поле №2).
- <Запас в килограммах> – количество ингредиента, натуральное число (поле №3).
- <Текстовая заметка> – текстовая заметка об ингредиенте, 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 Морковка оранжевый сладкий 100 ДляСупов	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ Морковка
ВВЕСТИ Мука белая сладкая 500 ДляПирожков	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ Мука ДОБАВЛЕНА НОВАЯ ЗАПИСЬ
ВВЕСТИ Виноград красный кислый 10 ДляСалатов	Виноград Виноград красный кислый 10
СОРТИРОВАТЬ 3	ДляСалатов
ИСКЛЮЧИТЬ Виноград	

<p>ОТОБРАЗИТЬ Морковка СОЗДАТЬ_ФАЙЛ kitchen.txt ИСКЛЮЧИТЬ Мука СОРТИРОВАТЬ 1 ИМПОРТ kitchen.txt СОРТИРОВАТЬ 0</p>	<p>Морковка оранжевый сладкий 100 ДляСупов Мука белая сладкая 500 ДляПирожков ДАННЫЕ ОТСОРТИРОВАНЫ ПО 3 ЗАПИСЬ Виноград УДАЛЕНА Морковка оранжевый сладкий 100 ДляСупов ИНФОРМАЦИЯ О ЗАПИСИ Морковка ПРЕДСТАВЛЕНА ВЫШЕ kitchen.txt ЗАПИСАН ЗАПИСЬ Мука УДАЛЕНА Морковка оранжевый сладкий 100 ДляСупов ДАННЫЕ ОТСОРТИРОВАНЫ ПО 1 kitchen.txt ЗАГРУЖЕН Морковка оранжевый сладкий 100 ДляСупов Мука белая сладкая 500 ДляПирожков ДАННЫЕ ОТСОРТИРОВАНЫ ПО 0</p>
---	--

### 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 не корректно 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	ОТОБРАЗИТЬ	5 баллов	3 балла	0 баллов
3	ИСКЛЮЧИТЬ	5 баллов	3 балла	0 баллов
4	СОРТИРОВАТЬ	5 баллов	3 балла	0 баллов
5	СОЗДАТЬ_ФАЙЛ	5 баллов	3 балла	0 баллов
6	ИМПОРТ	5 баллов	3 балла	0 баллов
<b>ИТОГО</b>		<b>30 баллов</b>	<b>18 баллов</b>	<b>0 баллов</b>

## Кейс №2.

### *Задание 1*

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

### *Задание 2*

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

### *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. Экпортируйте сборку в формат .stl. – 5 баллов.

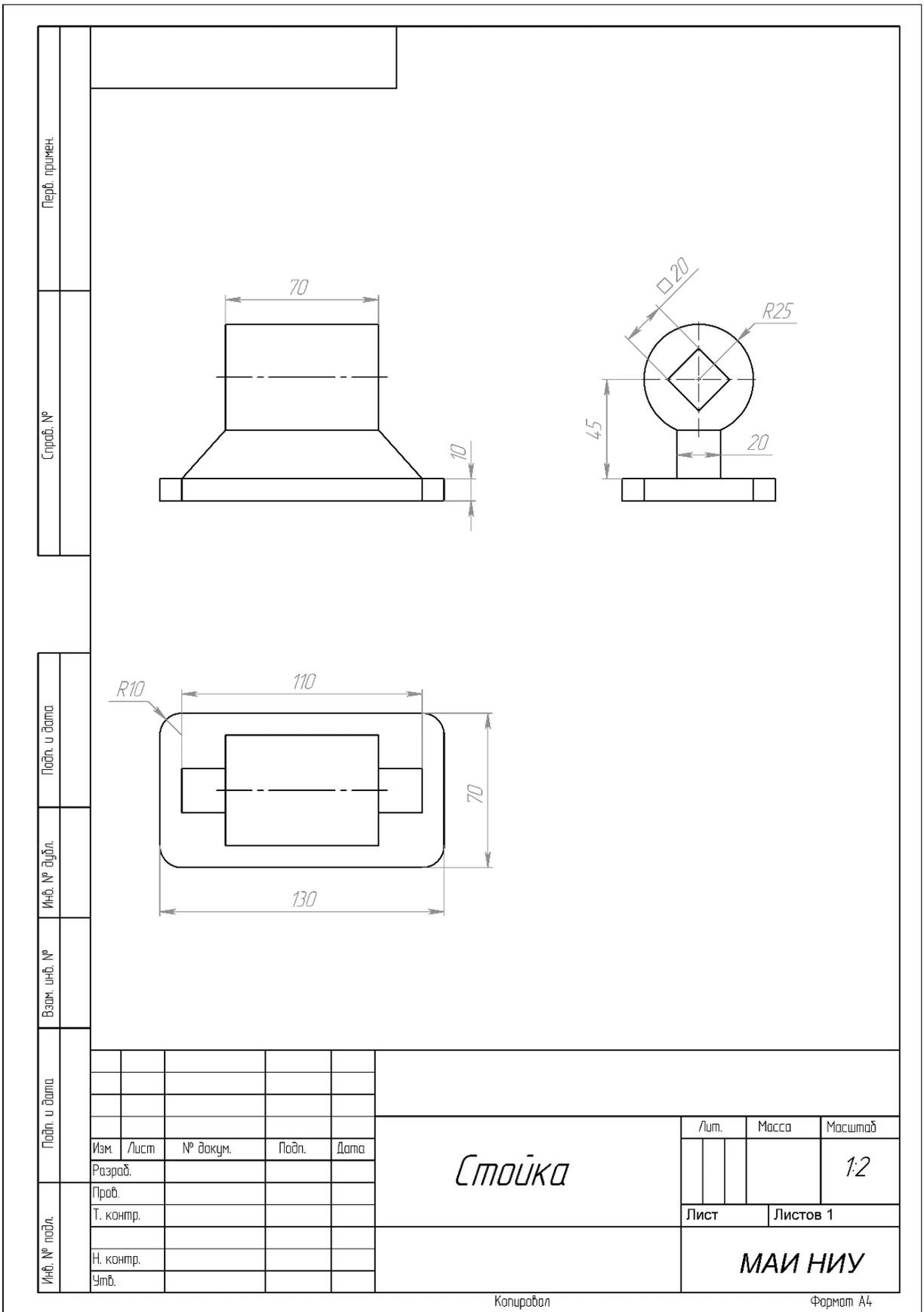


Рисунок 1

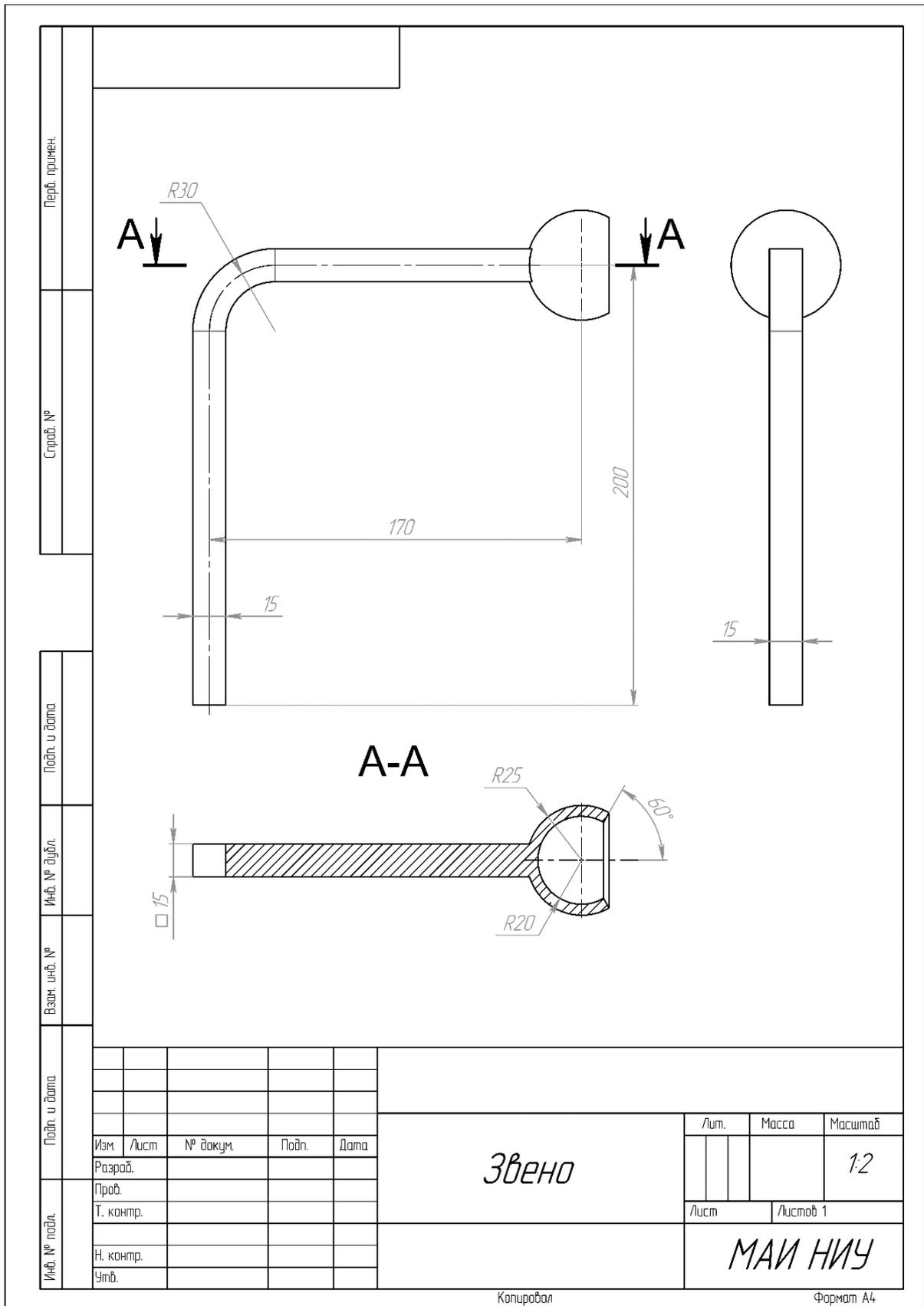


Рисунок 2



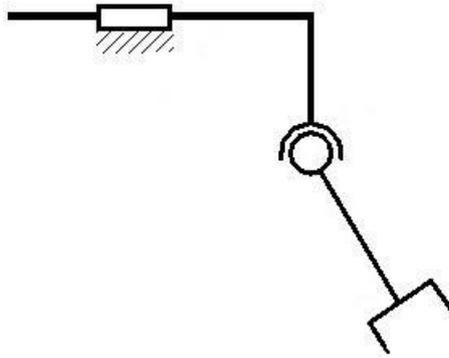


Рисунок 4

## Вариант 14

### Кейс №1.

#### «Программа учёта прохождения диспансеризации»

Разработать на языке Python программу с консольным интерфейсом для учёта результатов прохождения диспансеризации учеников. Данные о каждом ученике представляют собой совокупность следующих значений:

Номер	0	1	2	3	4
Поле	ФамилияИО	Возраст	Номер Школы	Специализация врача	Запись о результате осмотра

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:

1. **ВПИСАТЬ** <ФамилияИО> <Возраст> <Номер школы> <Специализация врача> <Запись о результате осмотра> – добавляет данные о новом ученике с заданными значениями. После выполнения функции в консоль приложения записывается фраза: **ДОБАВЛЕНА НОВАЯ ЗАПИСЬ** <Название>
2. **НАЙТИ** < ФамилияИО > – выводит в консоль всю информацию об ученике с заданной фамилией. После выполнения функции в консоль приложения записывается фраза: **ИНФОРМАЦИЯ О ЗАПИСИ** <Название> **ПРЕДСТАВЛЕНА ВЫШЕ**

3. СТЕРЕТЬ < ФамилияИО > – удаляет ученика с заданной фамилией. После выполнения функции в консоль приложения записывается фраза: ЗАПИСЬ <Название> УДАЛЕНА
4. УПОРЯДОЧИТЬ <номер\_поля\_сорт> – выводит в консоль информацию обо всех добавленных учениках, 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>.
5. ЭКСПОРТ <имя\_файла> – сохраняет добавленных учеников в заданный текстовый файл в формате команды ВПИСАТЬ (с разделением значений пробелами): <ФамилияИО> <Возраст> <Номер школы> <Специализация врача> <Запись о результате осмотра>. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> ЗАПИСАН.
6. СЧИТАТЬ <имя\_файла> – загружает данные об учениках из заданного текстового файла, полностью замещая текущий состав записей об учениках. Считается, что загружаемый файл был создан при вызове команды ЭКСПОРТ. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> ЗАГРУЖЕН.

Описание параметров команд:

- <ФамилияИО> – фамилия ученика, БЕЗ разделения пробелами (поле №0), например, КарповДИ. Данное значение является уникальным для каждого результата.
- <Возраст> – классификация ученика, натуральное число (поле №1).
- <Номер Школы> – классификация ученика, натуральное число (поле №2).
- <Специализация врача> – врач, проводивший осмотр ученику, строковое значение, не содержащее пробелов (поле №3).
- <Запись о результате осмотра> – текстовая заметка об осмотре, 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
ВПИСАТЬ ЗапорожскихАИ 15 2 Дерматолог Здоров ВПИСАТЬ БеловаНИ 17 5 Окулист ПрийтиНаПовторное	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ ЗапорожскихАИ ДОБАВЛЕНА НОВАЯ ЗАПИСЬ БеловаНИ

<p>ВПИСАТЬ ИвановАБ 13 12 Терапевт Здоров</p> <p>УПОРЯДОЧИТЬ 1 СТЕРЕТЬ ИвановАБ НАЙТИ БеловаНИ ЭКСПОРТ pupils.txt СТЕРЕТЬ БеловаНИ УПОРЯДОЧИТЬ 2 СЧИТАТЬ pupils.txt УПОРЯДОЧИТЬ 0</p>	<p>ДОБАВЛЕНА НОВАЯ ЗАПИСЬ ИвановаБ</p> <p>ИвановаБ 13 12 Терапевт Здоров</p> <p>ЗапорожскихАИ 15 2 Дерматолог Здоров</p> <p>БеловаНИ 17 5 Окулист ПрийтиНаПовторное</p> <p>ДАННЫЕ ОТСОРТИРОВАНЫ ПО 1 ЗАПИСЬ ИвановаБ УДАЛЕНА</p> <p>БеловаНИ 17 5 Окулист ПрийтиНаПовторное</p> <p>ИНФОРМАЦИЯ О ЗАПИСИ БеловаНИ ПРЕДСТАВЛЕНА ВЫШЕ pupils.txt ЗАПИСАН</p> <p>ЗАПИСЬ БеловаНИ УДАЛЕНА</p> <p>ЗапорожскихАИ 15 2 Дерматолог Здоров</p> <p>ДАННЫЕ ОТСОРТИРОВАНЫ ПО 2 pupils.txt ЗАГРУЖЕН</p> <p>БеловаНИ 17 5 Окулист ПрийтиНаПовторное</p> <p>ЗапорожскихАИ 15 2 Дерматолог Здоров</p> <p>ДАННЫЕ ОТСОРТИРОВАНЫ ПО 0</p>
---	--

### 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 не корректно при любом способе реализации – 0 баллов.

№ п/ п	Команда	Реализована как функция и	Реализована не как функция и	Выполняется не корректно при любом
--------------	---------	------------------------------	---------------------------------	--

		<b>выполняется корректно</b>	<b>выполняется корректно</b>	<b>способе реализации</b>
1	ВПИСАТЬ	5 баллов	3 балла	0 баллов
2	НАЙТИ	5 баллов	3 балла	0 баллов
3	СТЕРЕТЬ	5 баллов	3 балла	0 баллов
4	УПОРЯДОЧИТЬ	5 баллов	3 балла	0 баллов
	Ь			
5	ЭКСПОРТ	5 баллов	3 балла	0 баллов
6	СЧИТАТЬ	5 баллов	3 балла	0 баллов
<b>ИТОГО</b>		<b>30 баллов</b>	<b>18 баллов</b>	<b>0 баллов</b>

## Кейс №2.

### *Задание 1*

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

### *Задание 2*

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

### *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. Экпортируйте сборку в формат .stl. – 5 баллов.

Перб. примен.																																																																												
Спроб. №																																																																												
Подп. и дата																																																																												
Инд. № дубл.		Взам. инд. №		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%;"></td> </tr> <tr> <td style="text-align: center;">Изм.</td> <td style="text-align: center;">Лист</td> <td style="text-align: center;">№ докум.</td> <td style="text-align: center;">Подп.</td> <td style="text-align: center;">Дата</td> <td colspan="6" rowspan="3" style="text-align: center; vertical-align: middle; font-size: 2em;">Стойка</td> <td style="text-align: center;">Лит.</td> <td style="text-align: center;">Масса</td> <td style="text-align: center;">Масштаб</td> </tr> <tr> <td style="text-align: center;">Разраб.</td> <td></td> <td></td> <td></td> <td></td> <td style="text-align: center;">12</td> <td style="text-align: center;">12</td> <td style="text-align: center;">1:2</td> </tr> <tr> <td style="text-align: center;">Проб.</td> <td></td> <td></td> <td></td> <td></td> <td style="text-align: center;">Лист</td> <td style="text-align: center;">Листов 1</td> </tr> <tr> <td style="text-align: center;">Т. контр.</td> <td></td> <td></td> <td></td> <td></td> <td colspan="6" rowspan="3" style="text-align: center; vertical-align: middle; font-size: 1.5em;">МАИ НИУ</td> <td colspan="3"></td> </tr> <tr> <td style="text-align: center;">Н. контр.</td> <td></td> <td></td> <td></td> <td></td> <td colspan="3"></td> </tr> <tr> <td style="text-align: center;">Утб.</td> <td></td> <td></td> <td></td> <td></td> <td colspan="3"></td> </tr> </table>														Изм.	Лист	№ докум.	Подп.	Дата	Стойка						Лит.	Масса	Масштаб	Разраб.					12	12	1:2	Проб.					Лист	Листов 1	Т. контр.					МАИ НИУ									Н. контр.								Утб.							
Изм.	Лист	№ докум.	Подп.			Дата	Стойка						Лит.	Масса	Масштаб																																																													
Разраб.													12	12	1:2																																																													
Проб.													Лист	Листов 1																																																														
Т. контр.					МАИ НИУ																																																																							
Н. контр.																																																																												
Утб.																																																																												
Инд. № подл.		Подп. и дата		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%;"></td> </tr> <tr> <td style="text-align: center;">Изм.</td> <td style="text-align: center;">Лист</td> <td style="text-align: center;">№ докум.</td> <td style="text-align: center;">Подп.</td> <td style="text-align: center;">Дата</td> <td colspan="6" rowspan="3" style="text-align: center; vertical-align: middle; font-size: 2em;">Стойка</td> <td style="text-align: center;">Лит.</td> <td style="text-align: center;">Масса</td> <td style="text-align: center;">Масштаб</td> </tr> <tr> <td style="text-align: center;">Разраб.</td> <td></td> <td></td> <td></td> <td></td> <td style="text-align: center;">12</td> <td style="text-align: center;">12</td> <td style="text-align: center;">1:2</td> </tr> <tr> <td style="text-align: center;">Проб.</td> <td></td> <td></td> <td></td> <td></td> <td style="text-align: center;">Лист</td> <td style="text-align: center;">Листов 1</td> </tr> <tr> <td style="text-align: center;">Т. контр.</td> <td></td> <td></td> <td></td> <td></td> <td colspan="6" rowspan="3" style="text-align: center; vertical-align: middle; font-size: 1.5em;">МАИ НИУ</td> <td colspan="3"></td> </tr> <tr> <td style="text-align: center;">Н. контр.</td> <td></td> <td></td> <td></td> <td></td> <td colspan="3"></td> </tr> <tr> <td style="text-align: center;">Утб.</td> <td></td> <td></td> <td></td> <td></td> <td colspan="3"></td> </tr> </table>														Изм.	Лист	№ докум.	Подп.	Дата	Стойка						Лит.	Масса	Масштаб	Разраб.					12	12	1:2	Проб.					Лист	Листов 1	Т. контр.					МАИ НИУ									Н. контр.								Утб.							
Изм.	Лист	№ докум.	Подп.			Дата	Стойка						Лит.	Масса	Масштаб																																																													
Разраб.													12	12	1:2																																																													
Проб.													Лист	Листов 1																																																														
Т. контр.					МАИ НИУ																																																																							
Н. контр.																																																																												
Утб.																																																																												
Инд. № подл.		Подп. и дата		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%;"></td> </tr> <tr> <td style="text-align: center;">Изм.</td> <td style="text-align: center;">Лист</td> <td style="text-align: center;">№ докум.</td> <td style="text-align: center;">Подп.</td> <td style="text-align: center;">Дата</td> <td colspan="6" rowspan="3" style="text-align: center; vertical-align: middle; font-size: 2em;">Стойка</td> <td style="text-align: center;">Лит.</td> <td style="text-align: center;">Масса</td> <td style="text-align: center;">Масштаб</td> </tr> <tr> <td style="text-align: center;">Разраб.</td> <td></td> <td></td> <td></td> <td></td> <td style="text-align: center;">12</td> <td style="text-align: center;">12</td> <td style="text-align: center;">1:2</td> </tr> <tr> <td style="text-align: center;">Проб.</td> <td></td> <td></td> <td></td> <td></td> <td style="text-align: center;">Лист</td> <td style="text-align: center;">Листов 1</td> </tr> <tr> <td style="text-align: center;">Т. контр.</td> <td></td> <td></td> <td></td> <td></td> <td colspan="6" rowspan="3" style="text-align: center; vertical-align: middle; font-size: 1.5em;">МАИ НИУ</td> <td colspan="3"></td> </tr> <tr> <td style="text-align: center;">Н. контр.</td> <td></td> <td></td> <td></td> <td></td> <td colspan="3"></td> </tr> <tr> <td style="text-align: center;">Утб.</td> <td></td> <td></td> <td></td> <td></td> <td colspan="3"></td> </tr> </table>														Изм.	Лист	№ докум.	Подп.	Дата	Стойка						Лит.	Масса	Масштаб	Разраб.					12	12	1:2	Проб.					Лист	Листов 1	Т. контр.					МАИ НИУ									Н. контр.								Утб.							
Изм.	Лист	№ докум.	Подп.			Дата	Стойка						Лит.	Масса	Масштаб																																																													
Разраб.													12	12	1:2																																																													
Проб.													Лист	Листов 1																																																														
Т. контр.					МАИ НИУ																																																																							
Н. контр.																																																																												
Утб.																																																																												
Инд. № подл.		Подп. и дата		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%;"></td> </tr> <tr> <td style="text-align: center;">Изм.</td> <td style="text-align: center;">Лист</td> <td style="text-align: center;">№ докум.</td> <td style="text-align: center;">Подп.</td> <td style="text-align: center;">Дата</td> <td colspan="6" rowspan="3" style="text-align: center; vertical-align: middle; font-size: 2em;">Стойка</td> <td style="text-align: center;">Лит.</td> <td style="text-align: center;">Масса</td> <td style="text-align: center;">Масштаб</td> </tr> <tr> <td style="text-align: center;">Разраб.</td> <td></td> <td></td> <td></td> <td></td> <td style="text-align: center;">12</td> <td style="text-align: center;">12</td> <td style="text-align: center;">1:2</td> </tr> <tr> <td style="text-align: center;">Проб.</td> <td></td> <td></td> <td></td> <td></td> <td style="text-align: center;">Лист</td> <td style="text-align: center;">Листов 1</td> </tr> <tr> <td style="text-align: center;">Т. контр.</td> <td></td> <td></td> <td></td> <td></td> <td colspan="6" rowspan="3" style="text-align: center; vertical-align: middle; font-size: 1.5em;">МАИ НИУ</td> <td colspan="3"></td> </tr> <tr> <td style="text-align: center;">Н. контр.</td> <td></td> <td></td> <td></td> <td></td> <td colspan="3"></td> </tr> <tr> <td style="text-align: center;">Утб.</td> <td></td> <td></td> <td></td> <td></td> <td colspan="3"></td> </tr> </table>														Изм.	Лист	№ докум.	Подп.	Дата	Стойка						Лит.	Масса	Масштаб	Разраб.					12	12	1:2	Проб.					Лист	Листов 1	Т. контр.					МАИ НИУ									Н. контр.								Утб.							
Изм.	Лист	№ докум.	Подп.			Дата	Стойка						Лит.	Масса	Масштаб																																																													
Разраб.													12	12	1:2																																																													
Проб.													Лист	Листов 1																																																														
Т. контр.					МАИ НИУ																																																																							
Н. контр.																																																																												
Утб.																																																																												
Инд. № подл.		Подп. и дата		<table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 5%;"></td> </tr> <tr> <td style="text-align: center;">Изм.</td> <td style="text-align: center;">Лист</td> <td style="text-align: center;">№ докум.</td> <td style="text-align: center;">Подп.</td> <td style="text-align: center;">Дата</td> <td colspan="6" rowspan="3" style="text-align: center; vertical-align: middle; font-size: 2em;">Стойка</td> <td style="text-align: center;">Лит.</td> <td style="text-align: center;">Масса</td> <td style="text-align: center;">Масштаб</td> </tr> <tr> <td style="text-align: center;">Разраб.</td> <td></td> <td></td> <td></td> <td></td> <td style="text-align: center;">12</td> <td style="text-align: center;">12</td> <td style="text-align: center;">1:2</td> </tr> <tr> <td style="text-align: center;">Проб.</td> <td></td> <td></td> <td></td> <td></td> <td style="text-align: center;">Лист</td> <td style="text-align: center;">Листов 1</td> </tr> <tr> <td style="text-align: center;">Т. контр.</td> <td></td> <td></td> <td></td> <td></td> <td colspan="6" rowspan="3" style="text-align: center; vertical-align: middle; font-size: 1.5em;">МАИ НИУ</td> <td colspan="3"></td> </tr> <tr> <td style="text-align: center;">Н. контр.</td> <td></td> <td></td> <td></td> <td></td> <td colspan="3"></td> </tr> <tr> <td style="text-align: center;">Утб.</td> <td></td> <td></td> <td></td> <td></td> <td colspan="3"></td> </tr> </table>														Изм.	Лист	№ докум.	Подп.	Дата	Стойка						Лит.	Масса	Масштаб	Разраб.					12	12	1:2	Проб.					Лист	Листов 1	Т. контр.					МАИ НИУ									Н. контр.								Утб.							
Изм.	Лист	№ докум.	Подп.			Дата	Стойка						Лит.	Масса	Масштаб																																																													
Разраб.													12	12	1:2																																																													
Проб.													Лист	Листов 1																																																														
Т. контр.					МАИ НИУ																																																																							
Н. контр.																																																																												
Утб.																																																																												

Копировал

Формат А4

Рисунок 1

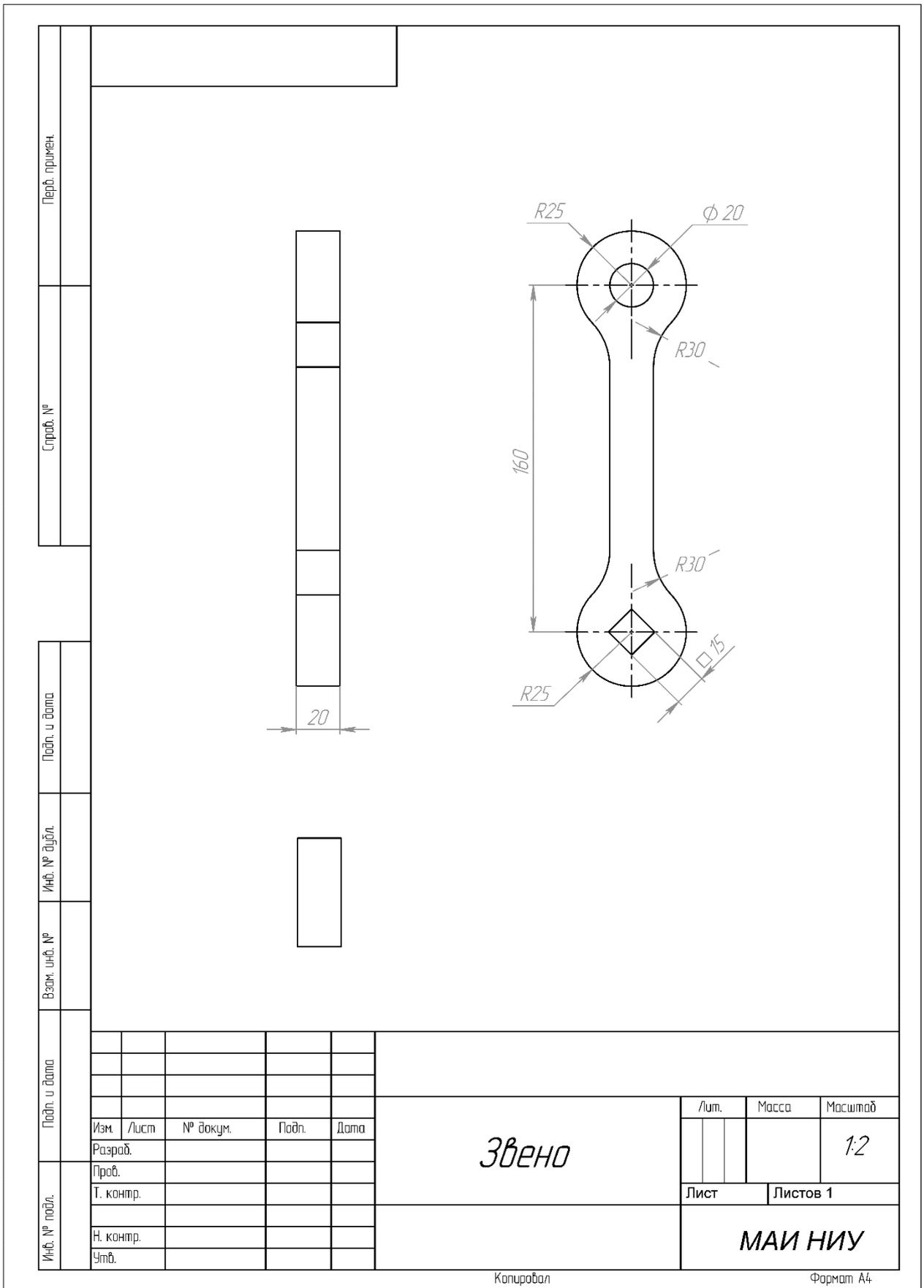


Рисунок 2

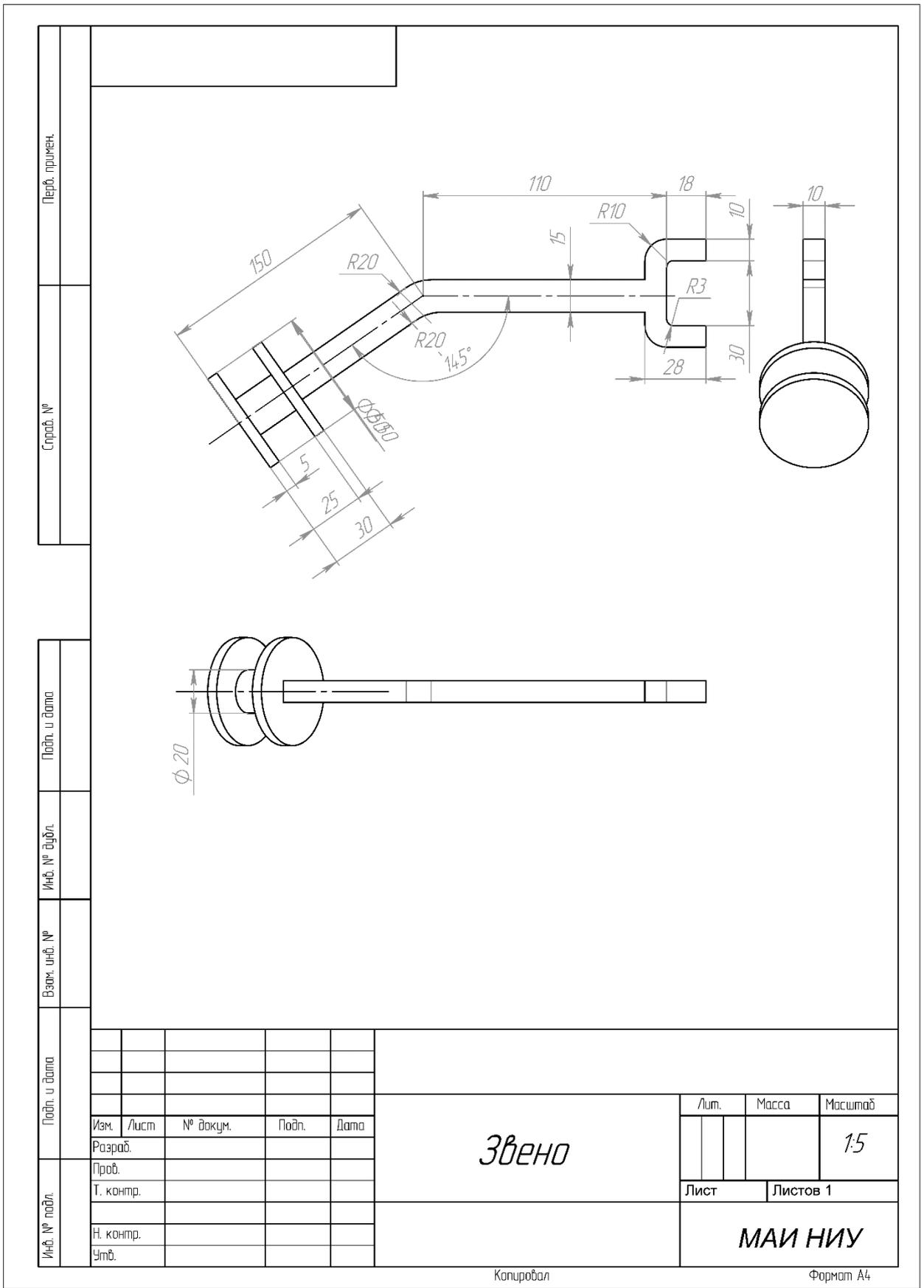


Рисунок 3

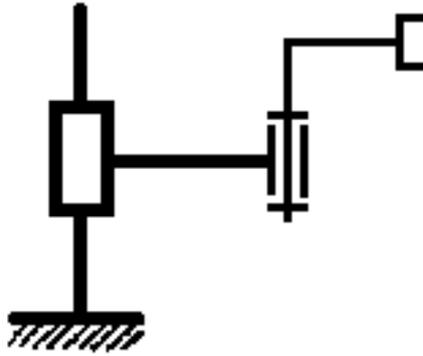


Рисунок 4

## Вариант 15

### Кейс №1.

«Программа учёта природных ресурсов государств»

Разработать на языке Python программу с консольным интерфейсом для учёта природных ресурсов государств. Данные о каждом природном ресурсе представляют собой совокупность следующих значений:

Номер	0	1	2	3	4
Поле	Название	Страна	Вид ресурса	Запас в триллионах тонн	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:

1. ЗАДАТЬ <Название> <Страна> <Вид ресурса> <Запаса в триллионах тонн> <Текстовая заметка> – добавляет данные о новом природном ресурсе с заданными значениями. После выполнения функции в консоль приложения записывается фраза: ДОБАВЛЕНА НОВАЯ ЗАПИСЬ <Название>
2. ВЫВЕСТИ <Название> – выводит в консоль всю информацию о природном ресурсе с заданным названием. После выполнения функции в консоль приложения записывается фраза: ИНФОРМАЦИЯ О ЗАПИСИ <Название> ПРЕДСТАВЛЕНА ВЫШЕ

3. УБРАТЬ <Название> – удаляет природный ресурс с заданным названием. После выполнения функции в консоль приложения записывается фраза: ЗАПИСЬ <Название> УДАЛЕНА
4. СПИСОК <номер\_поля\_сорт> – выводит в консоль информацию обо всех добавленных природных ресурсах, 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>.
5. ЗАПИСАТЬ <имя\_файла> – сохраняет добавленные природные ресурсы в заданный текстовый файл в формате команды ЗАДАТЬ (с разделением значений пробелами): <Название> <Страна> <Вид ресурса> <Запас в триллионах тонн> <Текстовая заметка>. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> ЗАПИСАН
6. ПРОЧИТАТЬ\_ФАЙЛ <имя\_файла> – загружает данные о природных ресурсах из заданного текстового файла, полностью замещая текущий состав записей о природных ресурсах. 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 природного ресурса, БЕЗ разделения пробелами (поле №0), например, ЖелезнаяРуда. Данное значение является уникальным для каждого ресурса.
- <Страна> – место добычи природного ресурса, строковое значение, не содержащее пробелов (поле №1).
- <Вид ресурса> – классификация природного ресурса, строковое значение, не содержащее пробелов (поле №2).
- <Запас в триллионах тонн> – количество природного ресурса, натуральное число (поле №3).
- <Текстовая заметка> – текстовая заметка о природном ресурсе, 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
ЗАДАТЬ Леса Казахстан невозобновляемые 90 Смолистые ЗАДАТЬ Вода Китай возобновляемые 4035 НельзяПить ЗАДАТЬ Уголь Россия невозобновляемые 1253 Дешевый СПИСОК 3 УБРАТЬ Вода ВЫВЕСТИ Леса ЗАПИСАТЬ resources.txt УБРАТЬ Леса СПИСОК 2 ПРОЧИТАТЬ_ФАЙЛ resources.txt СПИСОК 0	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ Леса ДОБАВЛЕНА НОВАЯ ЗАПИСЬ Вода ДОБАВЛЕНА НОВАЯ ЗАПИСЬ Уголь Леса Казахстан невозобновляемые 90 Смолистые Уголь Россия невозобновляемые 1253 Дешевый Вода Китай возобновляемые 4035 НельзяПить ДАННЫЕ ОТСОРТИРОВАНЫ ПО 3 ЗАПИСЬ Вода УДАЛЕНА Леса Казахстан невозобновляемые 90 Смолистые ИНФОРМАЦИЯ О ЗАПИСИ Леса ПРЕДСТАВЛЕНА ВЫШЕ resources.txt ЗАПИСАН ЗАПИСЬ Леса УДАЛЕНА Уголь Россия невозобновляемые 1253 Дешевый ДАННЫЕ ОТСОРТИРОВАНЫ ПО 1 resources.txt ЗАГРУЖЕН Леса Казахстан невозобновляемые 90 Смолистые Уголь Россия невозобновляемые 1253 Дешевый ДАННЫЕ ОТСОРТИРОВАНЫ ПО 0

#### 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 не корректно 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	ЗАДАТЬ	5 баллов	3 балла	0 баллов
2	ВЫВЕСТИ	5 баллов	3 балла	0 баллов
3	УБРАТЬ	5 баллов	3 балла	0 баллов
4	СПИСОК	5 баллов	3 балла	0 баллов
5	ЗАПИСАТЬ	5 баллов	3 балла	0 баллов
6	ПРОЧИТАТЬ_ФАЙЛ	5 баллов	3 балла	0 баллов
<b>ИТОГО</b>		<b>30 баллов</b>	<b>18 баллов</b>	<b>0 баллов</b>

## Кейс №2.

### *Задание 1*

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

### *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.

### *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.

Перв. примен.		
Спраб. №		
Подп. и дата	Инд. № одбл.	
Взам. инд. №		
Подп. и дата		
Инд. № подл.		

Изм.	Лист	№ докум.	Подп.	Дата	<i>Основание</i>			Лит.	Масса	Масштаб		
Разраб.												1:2
Проб.								Лист	Листов 1			
Н. контр.					<i>МАИ НИУ</i>							
Утв.												

Копировал

Формат А4

Рисунок 1

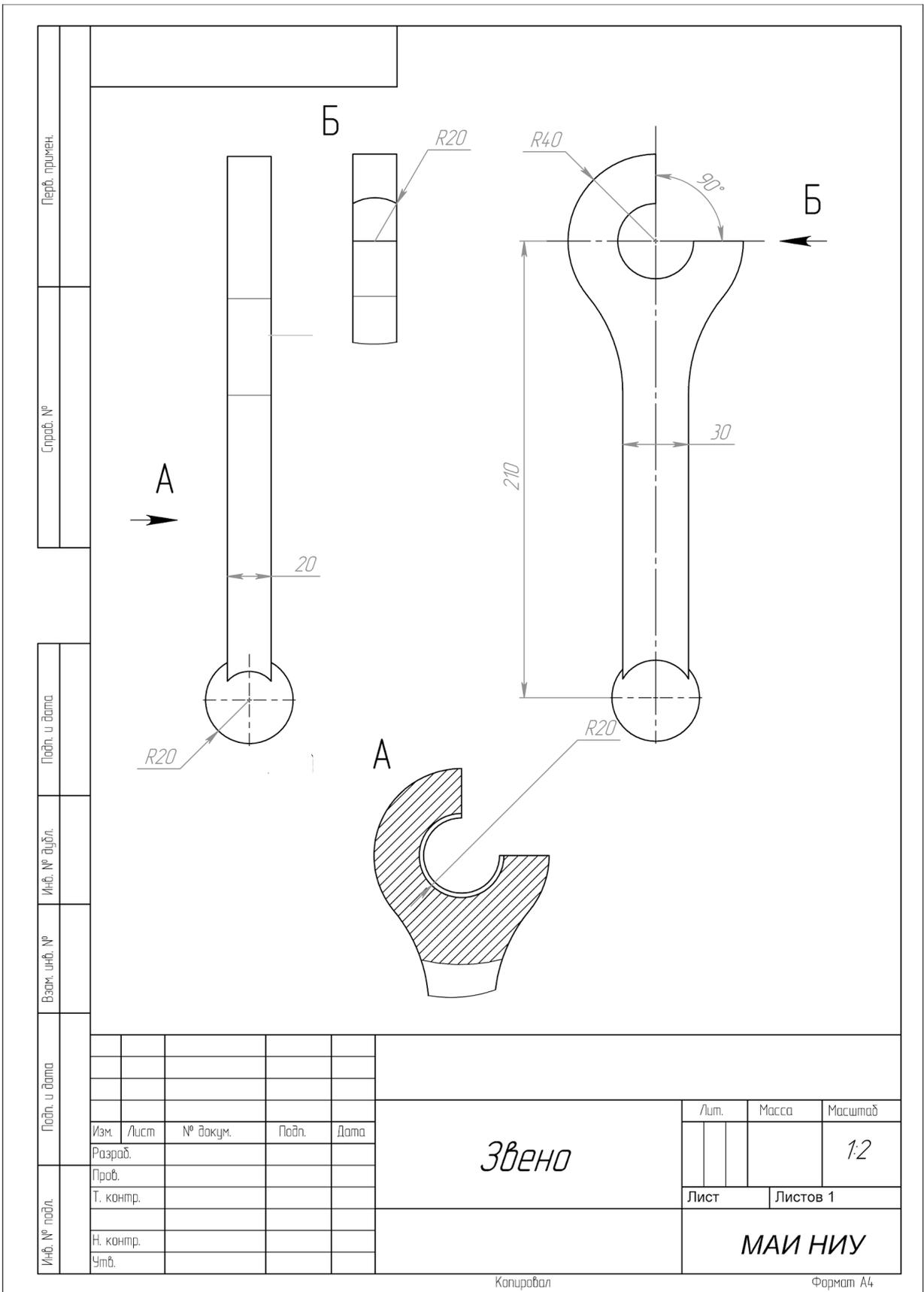


Рисунок 2

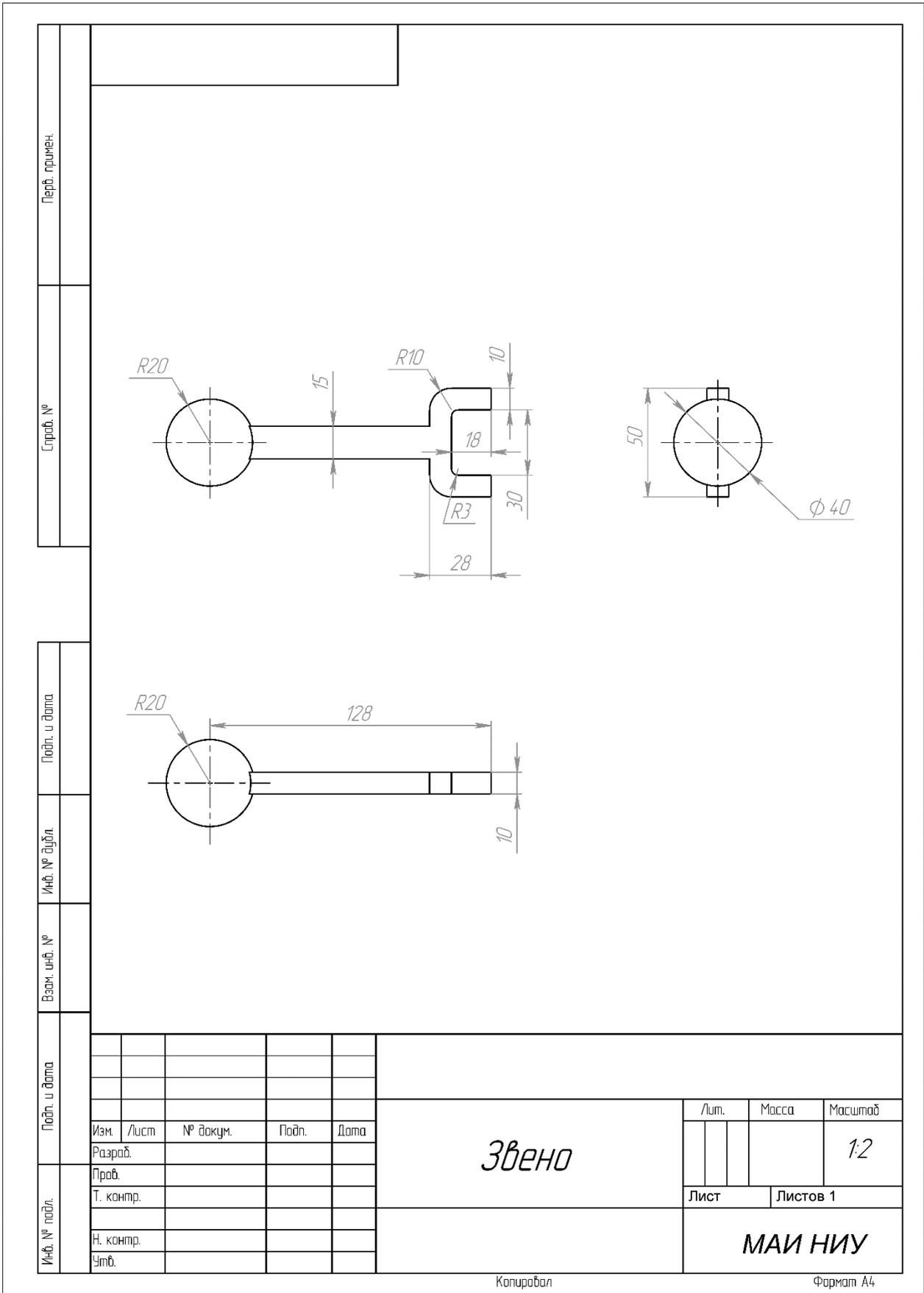


Рисунок 3

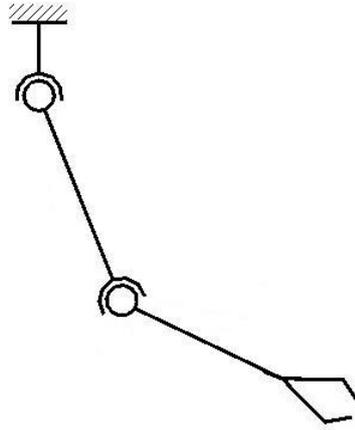


Рисунок 4

## Вариант 16

### Кейс №1.

«Программа учёта автомобилей в автосалоне»

Разработать на языке Python программу с консольным интерфейсом для учёта автомобилей. Данные о каждом автомобиле представляют собой совокупность следующих значений:

Номер	0	1	2	3	4
Поле	ФамилияИО	Марка	Цвет	Стоимость	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:

1. **ВВЕСТИ** <ФамилияИО> <Марка> <Цвет> <Стоимость> <Текстовая заметка> – добавляет данные о новом автомобиле с заданными значениями. После выполнения функции в консоль приложения записывается фраза: **ДОБАВЛЕНА НОВАЯ ЗАПИСЬ** <Название>
2. **ВЫВЕСТИ** <ФамилияИО> – выводит в консоль всю информацию об автомобиле с заданной фамилией владельца. После выполнения функции в консоль приложения записывается фраза: **ИНФОРМАЦИЯ О ЗАПИСИ** <Название> **ПРЕДСТАВЛЕНА ВЫШЕ**
3. **СТЕРЕТЬ** <ФамилияИО> – удаляет автомобиль с заданной фамилией владельца. После выполнения функции в консоль приложения записывается фраза: **ЗАПИСЬ** <Название> **УДАЛЕНА**

4. СОРТИРОВАТЬ <номер\_поля\_сорт> – выводит в консоль информацию обо всех добавленных автомобилях, 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>.
5. ЗАПИСАТЬ <имя\_файла> – сохраняет добавленные автомобили в заданный текстовый файл в формате команды ВВЕСТИ (с разделением значений пробелами): <ФамилияИО> <Марка> <Цвет> <Стоимость> <Текстовая заметка>. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> ЗАПИСАН.
6. СЧИТАТЬ <имя\_файла> – загружает данные об автомобиле из заданного текстового файла, полностью замещая текущий состав записей об автомобилях. Считается, что загружаемый файл был создан при вызове команды ЗАПИСАТЬ. После выполнения функции в консоль приложения записывается фраза: <имя\_файла> ЗАГРУЖЕН.

Описание параметров команд:

- <ФамилияИО> – фамилия владельца автомобиля, БЕЗ разделения пробелами (поле №0), например, АкатьевЯВ. Данное значение является уникальным для каждого автомобиля.
- <Марка> – классификация автомобиля, строковое значение, не содержащее пробелов (поле №1).
- <Цвет> – характеристика автомобиля, строковое значение, не содержащее пробелов (поле №2).
- <Стоимость> – характеристика автомобиля, натуральное число (поле №3).
- <Текстовая заметка> – текстовая заметка об автомобиле, 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 АлишерБГ УАЗ Зеленый 1000000 ОтечАвтоПром	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ АлишерБГ
ВВЕСТИ НиколаевЛД КАМАЗ Оранжевый 3240000 ОтечАвтоПром	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ НиколаевЛД
ВВЕСТИ УженковДИ Рено белая 1433944 Битая	ДОБАВЛЕНА НОВАЯ ЗАПИСЬ УженковДИ

<p>СОРТИРОВАТЬ 3  СТЕРЕТЬ АлишерБГ  ВЫВЕСТИ УженковДИ  ЗАПИСАТЬ auto.txt  СТЕРЕТЬ НиколаевЛД  СОРТИРОВАТЬ 2  СЧИТАТЬ auto.txt  СОРТИРОВАТЬ 0</p>	<p>АлишерБГ УАЗ Зеленый 1000000  ОтечАвтоПром</p> <p>УженковДИ Рено белая 1433944 Битая</p> <p>НиколаевЛД КАМАЗ Оранжевый  3240000 ОтечАвтоПром</p> <p>ДАнные ОТСОРТИРОВАНЫ ПО 3  ЗАПИСЬ АлишерБГ УДАЛЕНА</p> <p>УженковДИ Рено белая 1433944 Битая</p> <p>ИНФОРМАЦИЯ О ЗАПИСИ  УженковДИ ПРЕДСТАВЛЕНА  ВЫШЕ</p> <p>auto.txt ЗАПИСАН</p> <p>ЗАПИСЬ НиколаевЛД УДАЛЕНА</p> <p>УженковДИ Рено белая 1433944 Битая</p> <p>ДАнные ОТСОРТИРОВАНЫ ПО 2  auto.txt ЗАГРУЖЕН</p> <p>УженковДИ Рено белая 1433944 Битая</p> <p>НиколаевЛД КАМАЗ Оранжевый  3240000 ОтечАвтоПром</p> <p>ДАнные ОТСОРТИРОВАНЫ ПО 0</p>
--	--

**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 не корректно 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
<b>ИТОГО</b>		<b>30 баллов</b>	<b>18 баллов</b>	<b>0 баллов</b>

## Кейс №2.

### *Задание 1*

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

### *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.

### *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.

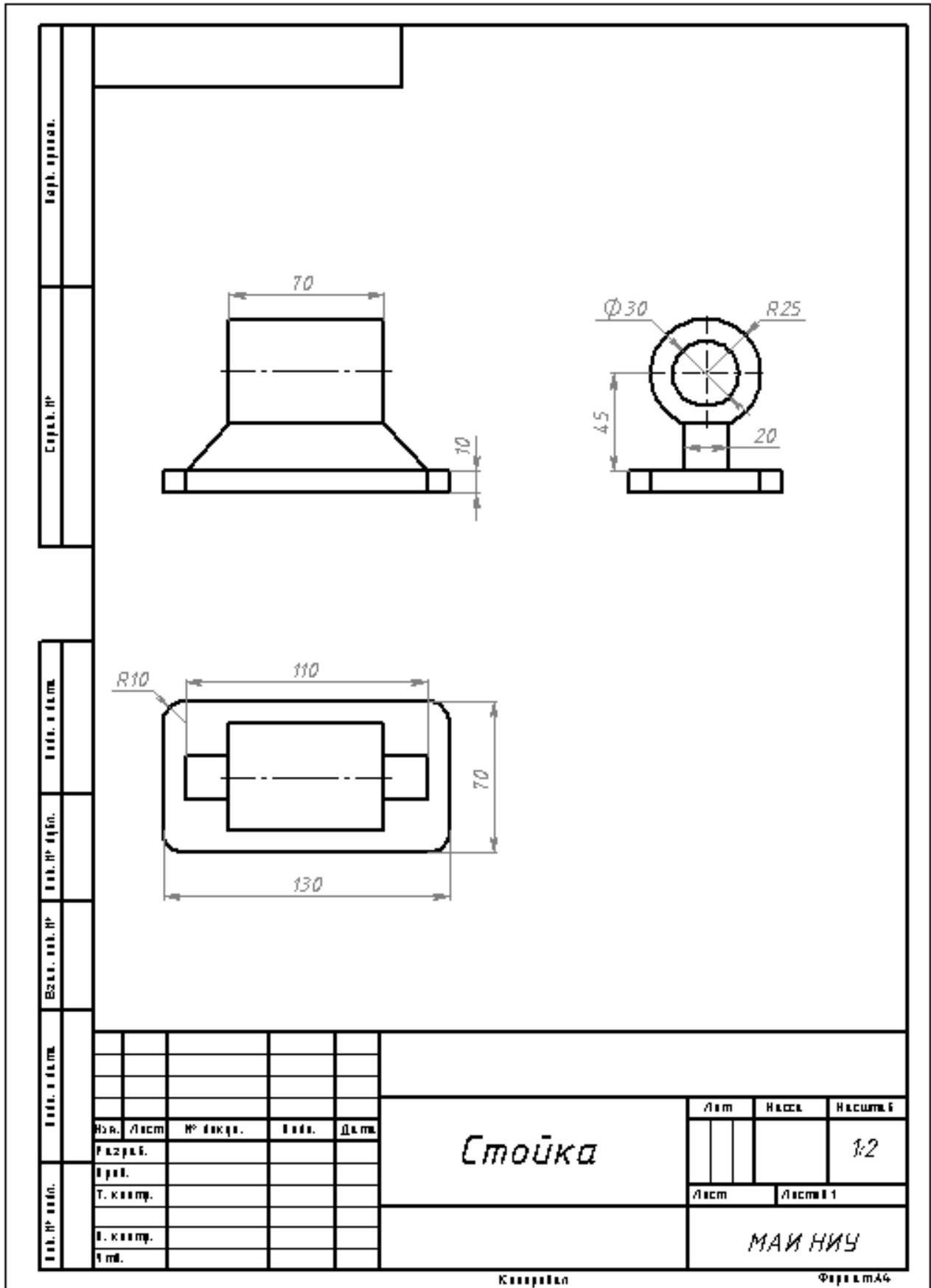


Рисунок 1

Перв. примен.													
Справ. №													
Подп. и дата													
Инф. № дубл.													
Взам. инв. №													
Подп. и дата													
Инв. № подл.													
Изм.					Лист	№ докум.	Подп.	Дата					
Т. контр.	<div style="font-size: 2em; font-weight: bold; margin-bottom: 10px;">Звено</div> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 20%; text-align: center;">Лит.</td> <td style="width: 20%; text-align: center;">Масса</td> <td style="width: 60%; text-align: center;">Масштаб</td> </tr> <tr> <td style="text-align: center;"> </td> <td style="text-align: center;"> </td> <td style="text-align: center;">1:2</td> </tr> <tr> <td colspan="2" style="text-align: center;">Лист</td> <td style="text-align: center;">Листов 1</td> </tr> </table>				Лит.	Масса	Масштаб			1:2	Лист		Листов 1
Лит.					Масса	Масштаб							
		1:2											
Лист		Листов 1											
Н. контр.	МАИ НИУ												
Утв.	Копировал <span style="float: right;">Формат А4</span>												

Рисунок 2

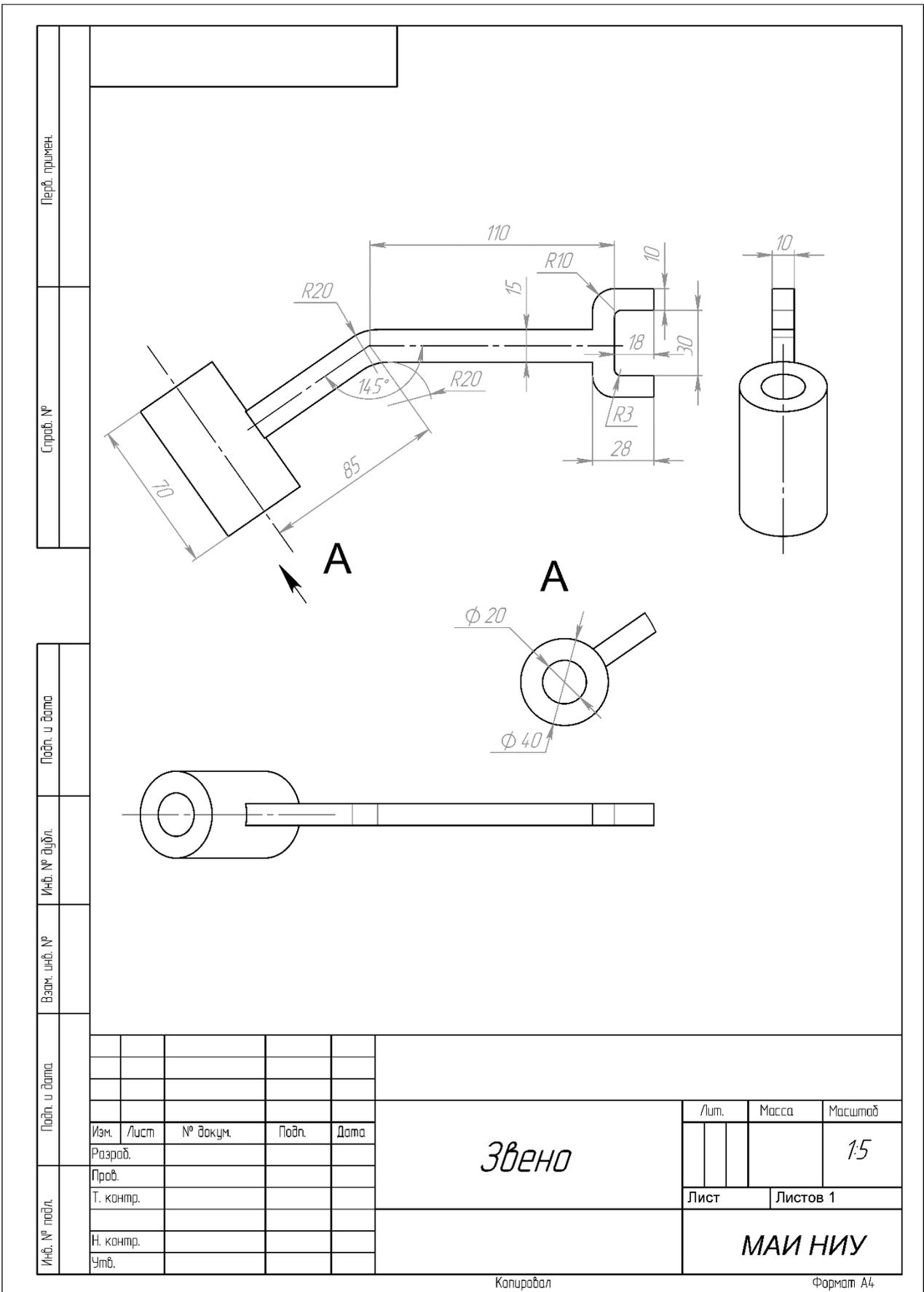


Рисунок 3

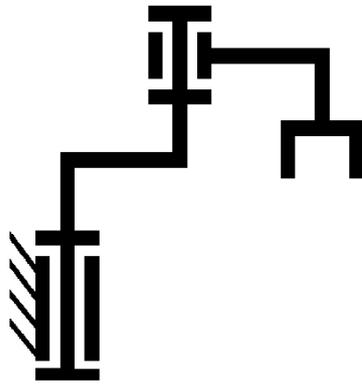


Рисунок 4