



## **Методические рекомендации**

**по способам решения заданий демонстрационного варианта, содержащие  
рекомендации по решению организационных вопросов, методику оценки  
заданий, описание возможных трудностей при подготовке, разбор типичных  
ошибок**

**Москва 2024**

# **Спецификация конкурсных материалов для проведения практического этапа Московского конкурса межпредметных навыков и знаний «Интеллектуальный мегаполис. Потенциал» в номинации «Инженерный класс» по направлению «Авиастроительные классы»**

## **1. Назначение конкурсных материалов**

Материалы практического этапа Московского конкурса межпредметных навыков и знаний «Интеллектуальный мегаполис. Потенциал» (далее – Конкурс) предназначены для оценки уровня практической подготовки участников Конкурса.

## **2. Условия проведения**

Практический этап Конкурса проводится в очном дистанционном формате с использованием технологии прокторинга. Участникам необходимо иметь компьютер (ПК или ноутбук) с выходом в Интернет, веб-камерой и микрофоном, а также смартфон (или планшет) со стабильным интернетом и приложением для считывания QR-кодов. Прохождение диагностики на мобильных устройствах невозможно. Требуется предварительная настройка оборудования. Браузер разрешается использовать только для просмотра заданий практического этапа, внесения ответов (загрузки файлов с решениями) и прохождения процедуры прокторинга.

При выполнении работы обеспечивается строгое соблюдение порядка организации и проведения Конкурса.

Задания практического этапа Конкурса выполняются с использованием следующего программного обеспечения (ПО):

1. Для выполнения заданий по кейсу №1 («Программирование»): среды разработки Python 3.9 и выше, PyCharm (Wing или VSCode).

2. Для выполнения заданий по кейсу №2 («3D-моделирование и 3D-печать»): системы автоматизированного проектирования (САПР) на выбор – T-Flex CAD, DraftSight, SolidWorks, Autodesk Inventor, Fusion 360, Компас 3D.

## **3. Продолжительность выполнения**

На выполнение заданий практического этапа Конкурса отводится **90 минут**. Во время проведения мероприятия участник может выйти из зоны проведения мероприятия не более чем на 5 минут, предупредив проктора на камеру. Мероприятие не продлевается на время отсутствия участника.

## **4. Содержание и структура**

Индивидуальный вариант участника включает 2 независимых кейса, базирующихся на содержании элективных курсов «Программирование», «3D-моделирование и 3D-печать».

## **5. Система оценивания**

Максимальный балл за выполнение кейса – 60 баллов. Участник выбирает для решения только один кейс из двух. Для получения максимального балла за практический этап Конкурса необходимо верно выполнить все задания выбранного кейса.

## **6. Приложения**

1. План конкурсных материалов для проведения практического этапа Конкурса.
2. Демонстрационный вариант конкурсных заданий практического этапа Конкурса.

**План конкурсных материалов для проведения практического этапа Московского конкурса межпредметных навыков и знаний «Интеллектуальный мегаполис. Потенциал» в номинации «Инженерный класс» по направлению «Авиастроительные классы»**

<b>№ задания</b>	<b>Уровень сложности</b>	<b>Уникальные кодификаторы Конкурса</b>	<b>Контролируемые требования к проверяемым умениям</b>	<b>Балл</b>
<b>Кейс №1 «Программирование»</b>				
1.1	повышенный	8.1. Объявление функции 8.2. Определение функции 8.3. Аргументы и параметры 8.5. Вызов функции	Создание функции для добавления элемента в массив	16
1.2	повышенный	8.1. Объявление функции 8.2. Определение функции 8.3. Аргументы и параметры 8.5. Вызов функции	Создание функции для удаления элемента в массиве	12
1.3	повышенный	8.1. Объявление функции 8.2. Определение функции 8.3. Аргументы и параметры 8.5. Вызов функции	Создание функции для форматированного вывода массива	16
1.4	повышенный	8.1. Объявление функции 8.2. Определение функции 8.3. Аргументы и параметры 8.5. Вызов функции	Создание функции для поиска в массиве	16
<b>Сумма баллов</b>				<b>60</b>
<b>Кейс №2 «3D-моделирование и 3D-печать»</b>				
2.1	базовый	4.2. Создание модели по размерам	Построение 3D модели	20
2.2	базовый	4.2. Создание модели по размерам	Построение 3D модели	30
2.3	повышенный	5.3. Оптимизация моделей для 3D печати	Подготовка моделей для 3D печати	10
<b>Сумма баллов</b>				<b>60</b>

**Демонстрационный вариант конкурсных заданий практического этапа Московского конкурса межпредметных навыков и знаний «Интеллектуальный мегаполис. Потенциал» в номинации «Инженерный класс» по направлению «Авиастроительные классы»**

**Кейс №1 «Программирование»**

**«Программа управления складом интернет-магазина»**

Разработать на языке Python программу с консольным интерфейсом для системы управления складом интернет-магазина.

Склад управляется с помощью системы учёта, которая должна поддерживать работу с товарами, заказами и отслеживать остатки на складе. Ваша задача - разработать набор функций для автоматизации процесса управления складом.

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

**Таблица с функциями:**

<b>Функция</b>	<b>Параметры</b>	<b>Описание</b>	<b>Пример вызова функции</b>
add_item	название: <b>str</b> идентификационный номер: <b>int</b> количество: <b>int</b>	Добавляет товар на склад. Если товар с таким идентификационным номером уже существует, увеличивает его количество.	add_item("Телефон", 101, 10)
remove_item	идентификационный номер: <b>int</b>	Удаляет товар со склада по его идентификационному номеру. Если товара нет на складе, выводит сообщение об ошибке.	remove_item(101)
view_stock	Нет параметров	Выводит список всех товаров на складе с указанием количества каждого из них.	view_stock()
search_item	название товара: <b>str</b>	Ищет товар по названию и выводит его идентификационный номер и количество на складе.	search_item("Телефон")

## Хранение данных:

- Все данные о товарах хранятся в виде глобальной переменной в виде словаря, где ключом является уникальный идентификационный номер, а значением - кортеж, содержащий название товара и его количество.
- Идентификационный номер и количество являются положительными целыми числами.
- Название – данное строчное значение является уникальным для каждого товара на складе.
- Идентификационный номер – данное целое числовое значение является уникальным для каждого товара на складе. Это означает, что для каждого товара есть уникальная пара значений (Название - Идентификационный номер), дополнительные проверки реализовать не нужно.
- Указание типов входных параметров в явном виде необязательно.
- Предполагается, что программе на вход подаются только корректные (не вызывающие ошибок) последовательности команд.

## Пример хранения данных о товарах

```
inventory = {  
    101: ("Телефон", 10),  
    102: ("Ноутбук", 5)  
}
```

Интерфейс для работы с пользователем:

Интерфейс должен быть реализован в виде текстовых команд, вводимых пользователем. Примерная структура интерфейса:

```
def main():  
  
    while True:  
  
        print("Добро пожаловать в систему управления складом!")  
  
        print("1. Добавить товар на склад")  
  
        print("2. Удалить товар со склада")  
  
        print("3. Просмотреть остатки на складе")  
  
        print("4. Поиск товара")  
  
        print("5. Выйти")  
  
        choice = int(input("Выберите действие: "))
```

А дальше необходимо реализовать корректный ввод данных, а также вызов соответствующей функции и обработки её результата.

## **Примеры ввода и вывода программы**

### Пример 1: Добавление товара на склад

#### **Ввод:**

Выберите действие: 1

Введите название товара: Телевизор

Введите идентификационный номер товара: 2001

Введите количество товара: 10

#### **Ожидаемый результат:**

Товар 'Телевизор' добавлен успешно.

#### **Дополнительный ввод (если товар уже существует):**

Выберите действие: 1

Введите название товара: Телевизор

Введите идентификационный номер товара: 2001

Введите количество товара: 5

#### **Ожидаемый результат:**

Товар 'Телевизор' обновлен. Количество на складе: 15.

### Пример 2: Удаление товара со склада

#### **Ввод:**

Выберите действие: 2

Введите идентификационный номер товара: 2001

#### **Ожидаемый результат:**

Товар удален успешно.

#### **Если товара нет в системе:**

Выберите действие: 2

Введите идентификационный номер товара: 9999

#### **Ожидаемый результат:**

Ошибка: Товар с идентификационным номером 9999 не найден.

### Пример 3: Просмотр остатков на складе

#### **Ввод:**

Выберите действие: 3

### **Ожидаемый результат:**

Список товаров на складе:  
Идентификационный номер: 2001  
Название: Телевизор  
Количество: 12

### Пример 4: Поиск товара

#### **Ввод:**

Выберите действие: 4  
Введите название товара: Телевизор

#### **Ожидаемый результат:**

Идентификационный номер товара: 2001  
Количество на складе: 12

#### **Если товар не найден:**

Выберите действие: 4  
Введите название товара: Холодильник

#### **Ожидаемый результат:**

Ошибка: Товар 'Холодильник' не найден.

### **Решение задачи**

```
# Хранилища данных
inventory = {} # Хранилище для товаров

# Функция добавления товара
def add_item (name, product_id, quantity):
    if product_id in inventory:
        inventory[product_id] = (name, inventory [product_id][1] + quantity)
        print(f"Товар '{name}' обновлен. Количество на складе {inventory
[product_id][1]}")
    else:
        inventory[product_id] = (name, quantity)
        print(f"Товар '{name}' добавлен успешно" )

# Функция удаления товара
def remove_item(product_id: int):
    if product_id in inventory:
        del inventory[product_id]
        print("Товар удален успешно.")
    else:
        print(f"Ошибка: Товар с идентификационным номером {product_id} не найден.")
```



```

# Функция просмотра остатков на складе
def view_stock():
    if not inventory:
        print("На складе нет товаров.")
        return
    print("Список товаров на складе:")
    for product_id, details in inventory.items():
        print(f"Идентификационный номер: {product_id}")
        print(f"Название: {details[0]}")
        print(f"Количество: {details[1]}\n")

# Функция поиска товара
def search_item(name: str):
    found = False
    for product_id, details in inventory.items():
        if details[0].lower() == name.lower():
            print(f"Идентификационный номер товара: {product_id}")
            print(f"Количество на складе: {details[1]}")
            found = True
            break
    if not found:
        print(f"Ошибка: Товар '{name}' не найден.")

def main():
    while True:
        print("\nДобро пожаловать в систему управления складом!")
        print("1. Добавить товар на склад")
        print("2. Удалить товар со склада")
        print("3. Просмотреть остатки на складе")
        print("4. Поиск товара")
        print("5. Выйти")

        choice = int(input("Выберите действие: "))

        if choice == 1:
            name = input("Введите название товара: ")
            id_ = int(input("Введите идентификационный номер товара: "))
            quantity = int(input("Введите количество товара: "))
            add_item(name, id_, quantity)
        elif choice == 2:
            id_ = int(input("Введите идентификационный номер товара: "))
            remove_item(id_)
        elif choice == 3:
            view_stock()
        elif choice == 4:
            name = input("Введите название товара: ")
            search_item(name)
        elif choice == 5:

```

```
        print("Выход из программы.")
        break
    else:
        print("Неверный выбор. Попробуйте снова.")

if __name__ == '__main__':
    main()
```

## **Методические рекомендации по решению задачи «Программа управления складом интернет-магазина»**

### **1. Общие положения**

Задача подразумевает разработку консольной программы на Python, которая будет управлять складом интернет-магазина. Основной акцент сделан на использование словаря для хранения данных о товарах и реализации функций, выполняющих добавление, удаление, просмотр и поиск товаров.

#### **Цель задачи:**

- Разработать систему управления складом, которая может добавлять, удалять и искать товары, а также выводить остатки товаров на складе.
- Реализовать командный интерфейс для взаимодействия с пользователем.

#### **Ожидания от учащихся:**

1. Умение работать с глобальными переменными для хранения информации о товарах.
2. Взаимодействие с пользователем через ввод и вывод.
3. Навыки работы со словарями и кортежами.
4. Реализация базовых операций с данными (добавление, удаление, поиск).
5. Обработка различных вариантов поведения программы при отсутствии или наличии товара.

### **2. Методика оценки задания**

#### **Критерии оценки:**

- Корректность работы программы:
- Программа должна корректно выполнять функции добавления, удаления, поиска и просмотра товаров.
- Реализация каждой функции должна быть точной и соответствовать требованиям задания.

#### **2. Обработка ошибок:**

- Реализована обработка ситуаций, когда товар не найден или его недостаточно для выполнения операции (например, при удалении).
- Сообщения об ошибках должны быть информативными и соответствовать требованиям задания.

### **3. Организация программы:**

- Логика программы разбита на функции, каждая из которых выполняет свою задачу.
- Программа не содержит лишнего кода и дублирования.
- Интерфейс пользователя должен быть удобным и понятным.

### **4. Соответствие интерфейса программе:**

- Программа должна соответствовать описанному интерфейсу взаимодействия с пользователем.
- Вывод данных должен соответствовать приведенным примерам.

### **3. Разбор возможных трудностей и типичных ошибок**

#### **Возможные трудности при подготовке:**

##### **1. Ошибки при работе со словарями:**

Ошибка при обращении к несуществующему ключу может вызвать исключение. Решение: проверка на существование товара перед выполнением операций (например, с использованием оператора `in`).

##### **2. Учет существующего товара при добавлении:**

Учащиеся могут забыть учесть ситуацию, когда товар с тем же идентификационным номером уже существует. В этом случае программа должна обновлять количество товара, а не добавлять новый товар.

#### **Типичные ошибки:**

##### **1. Дублирование товаров:**

Вместо обновления количества при добавлении товара, учащийся может допустить создание дублирующихся записей для одного товара. Решение: при добавлении товара необходимо проверять, существует ли он уже в базе данных.

##### **2. Некорректная работа функции удаления:**

Некоторые учащиеся могут не проверять, существует ли товар с указанным идентификационным номером, что приведет к ошибке. Решение: перед удалением проверять наличие товара в словаре.

##### **3. Ошибка при поиске товара по названию:**

Учащиеся могут неправильно реализовать поиск по названию, используя некорректные методы. Например, если учащийся попытается найти название как ключ словаря, программа выдаст ошибку, так как ключом является идентификационный номер.

##### **4. Отсутствие обработки случаев пустого склада:**

При просмотре остатков на складе, если нет товаров, программа может вывести пустой список или сообщение "Ошибка". Лучше предусмотреть вывод специального сообщения "Склад пуст", если товаров нет.

## **5. Отсутствие механизма выхода из программы:**

Иногда учащиеся забывают реализовать корректный выход из программы при выборе опции 5. Следует корректно завершать выполнение программы через `exit()` или `break` в цикле.

## **Кейс №2 «3D-моделирование и 3D-печать»**

### ***Задание 1.***

Постройте 3D-модель примитива согласно чертежу, представленному на рисунке 1. Экспортируйте детали в формат .stl.

### ***Задание 2.***

Постройте 3D-модель детали согласно чертежу, представленному на рисунке 2. Экспортируйте детали в формат .stl.

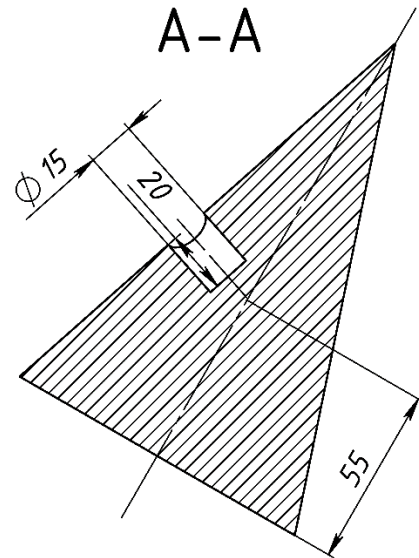
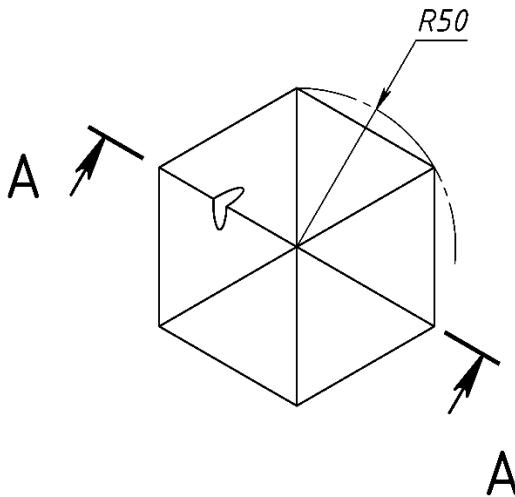
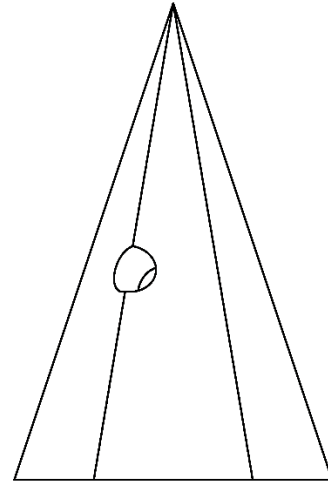
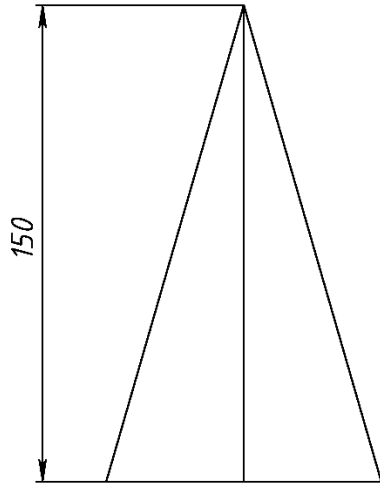
### ***Задание 3.***

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.

# Демовариант

Перв. примен.

Справ. №



Подп. и дата

Инв. № дубл.

Взам. инв. №

Подп. и дата

Инв. № подл.

Изм.	Лист	№ докум.	Подп.	Дата
Разраб.				
Пров.				
Т. контр.				
Н. контр.				
Чтв.				

## Демовариант

*Примитив*

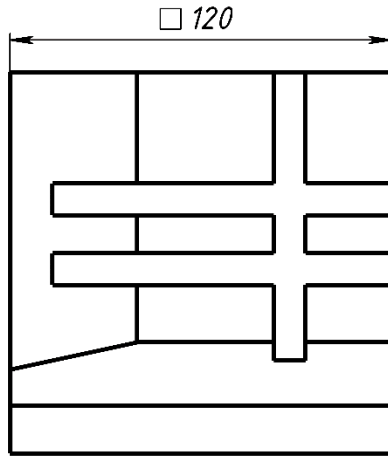
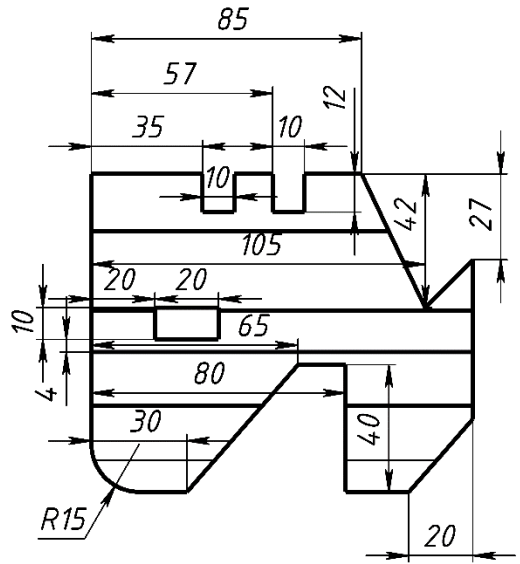
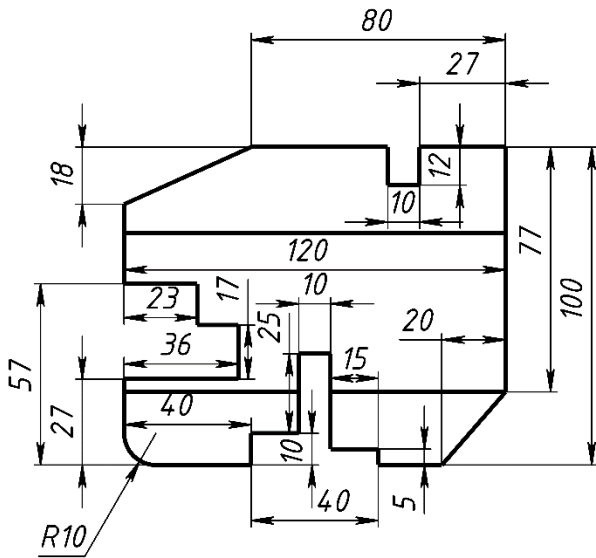
Лист	Масса	Масштаб
		1:2
Лист	Листов 1	
<b>МАИ НИУ</b>		

Копировал

Формат А4

Рисунок 1

Демовариант



Отверстие сквозное

Перв. примен.

Справ. №

Подп. и дата

Инв. № дубл.

Взам. инв. №

Подп. и дата

Инв. № подл.

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

Демовариант

Деталь

Лист	Масса	Масштаб
		1:2
Лист	Листов 1	
МАИ НИУ		

Копировал

Формат А4

Рисунок 2

### Критерии снижения оценки выполненных заданий

<b>Задание</b>	<b>Критерий</b>	<b>Количество снижаемых баллов</b>
2.1	Построенная деталь не соответствует конфигурации чертежа	5
2.1	Построенная деталь не соответствует размерам чертежа	5
2.2	Построенная деталь не соответствует конфигурации чертежа	10
2.2	Построенная деталь не соответствует размерам чертежа	5
2.3	За поверхность рабочего стола принтера принята негоризонтальная плоскость, отличная от плоскости Сверху. Тор, Вид сверху – в зависимости от выбранной САПР или детали пересекают данную плоскость	1
2.3	Расположение и ориентация деталей выбраны таким образом, что количество поддержек не минимально	2 (за каждую деталь)



**Пример выполнения:**

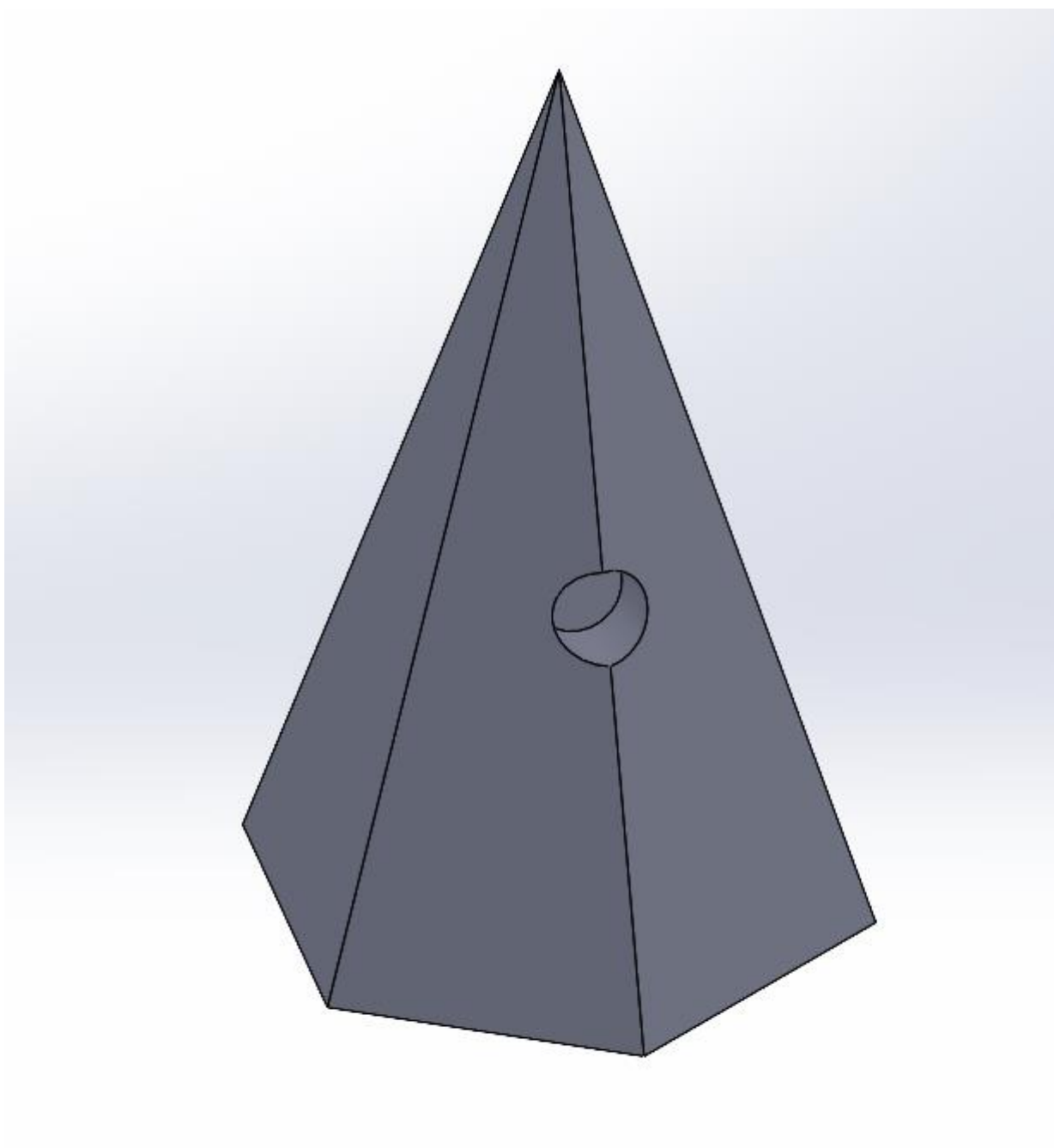


Рисунок 3 - Пример построения примитива

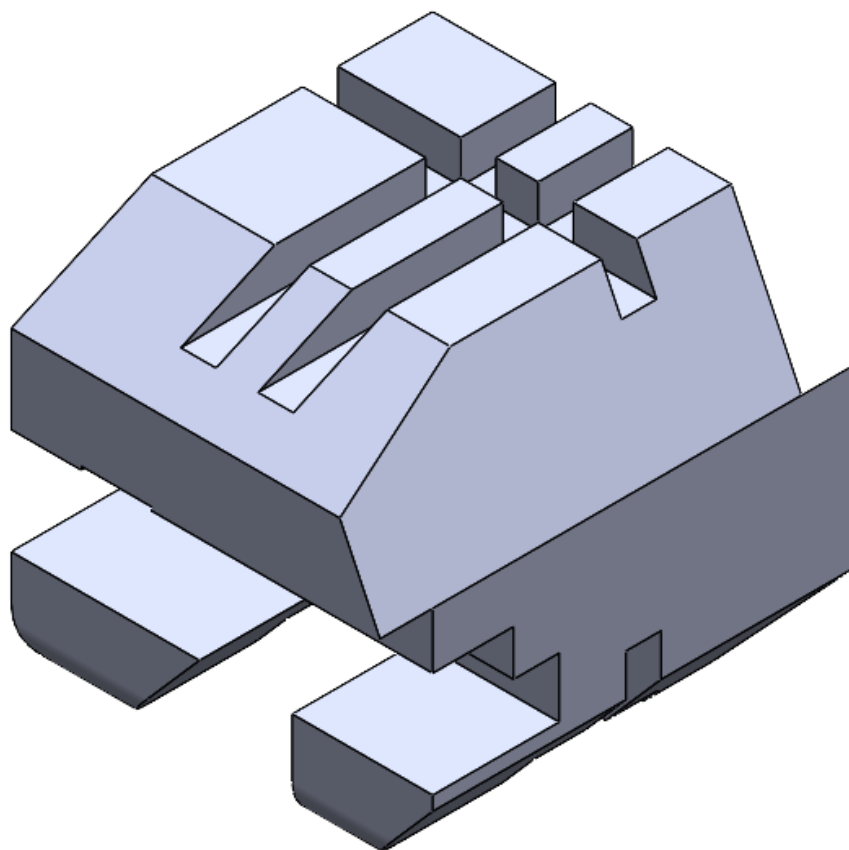


Рисунок 4 - Пример построения детали

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

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

Для построения деталей при выполнении экзаменационного задания с помощью САПР, следует придерживаться следующей последовательности действий:

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

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

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

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

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

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

Особое внимание требуется уделить построению 2D эскиза. Как правило, именно его размеры и геометрические взаимосвязи служат надежным инструментом при однозначном определении формы и размеров деталей. При простановке размеров эскиза следует использовать принцип «от базы», когда за базовую принимается, как правило, самая длинная линия в данном измерении и размеры, как правило, проставляют от нее. На рисунке 1 приведен пример простановки размеров «от базы» в горизонтальном направлении.

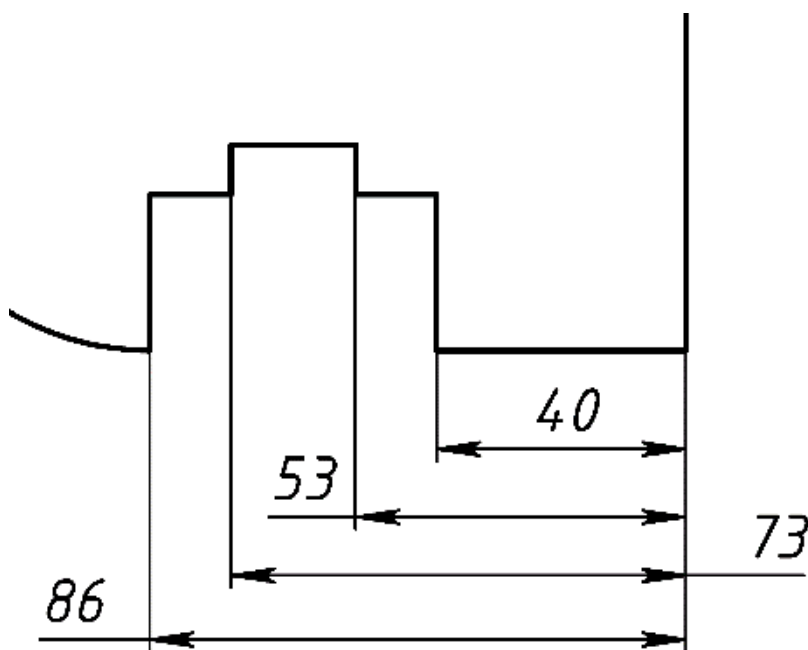


Рисунок 1

При построении эскиза рекомендуется сначала создать его примерную форму с помощью прямых линий и дуг, затем добавить геометрические взаимосвязи, уточнив его форму, а затем образмерить его с помощью размеров, указанных на чертежах.

При выполнении заданий на построение деталей, следует учитывать, что чертежи содержат полную информацию о форме и размерах деталей. При построении деталей возможна любая ориентация их в пространстве.

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

Указанная технология 3D печати подразумевает, что рабочее сопло принтера при печати каждого слоя обходит сначала по контуру слоя, а потом заполняет внутреннюю область этого контура сеткой. Это хорошо видно на недопечатанных деталях, как показано на рисунке 2.

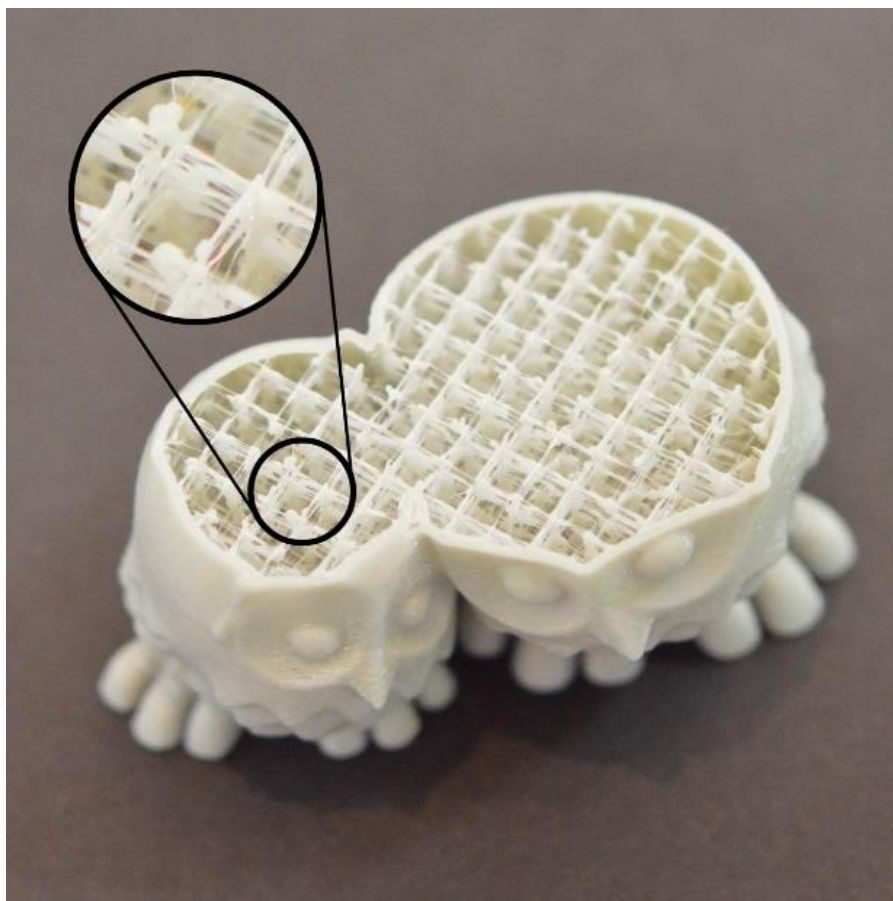


Рисунок 2

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

### **РЕЗУЛЬТАТ ВЫПОЛНЕНИЯ**

На рисунках 3-5 приведены примеры успешного выполнения заданий кейса №2.

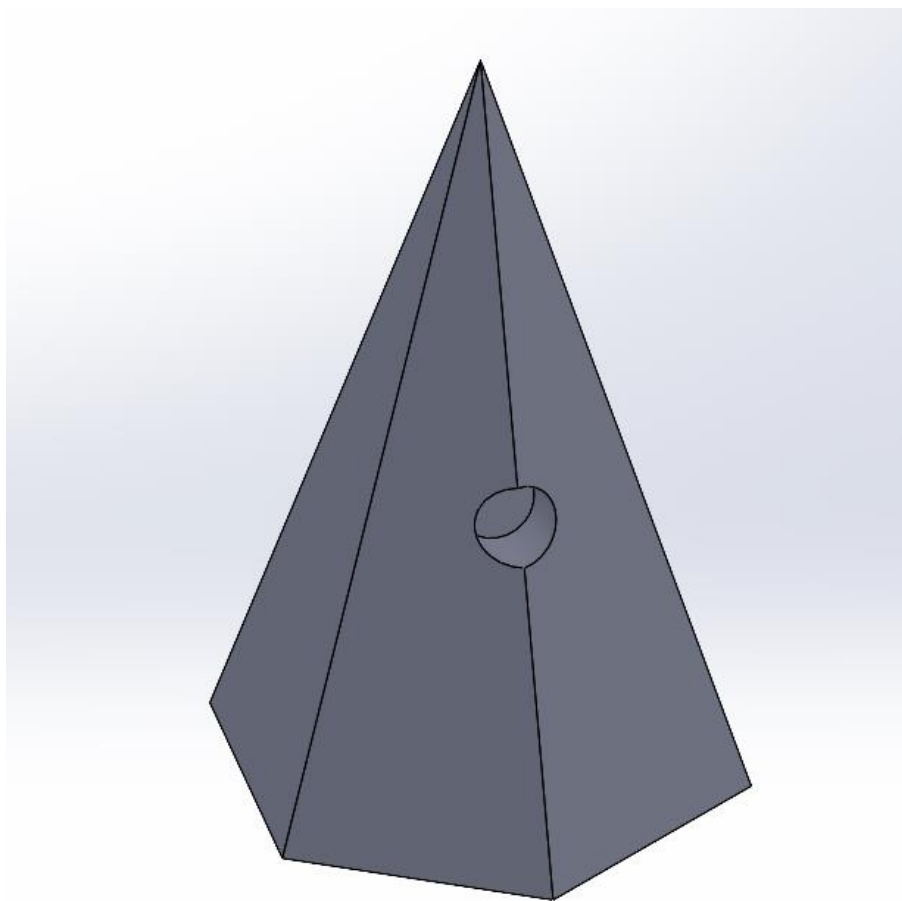


Рисунок 3

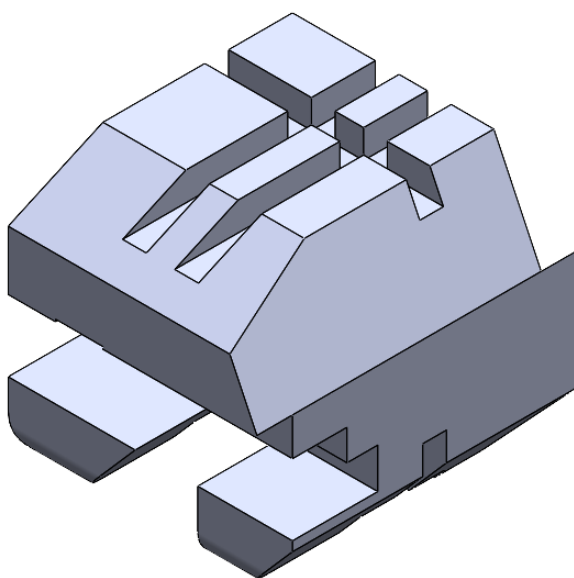


Рисунок 4

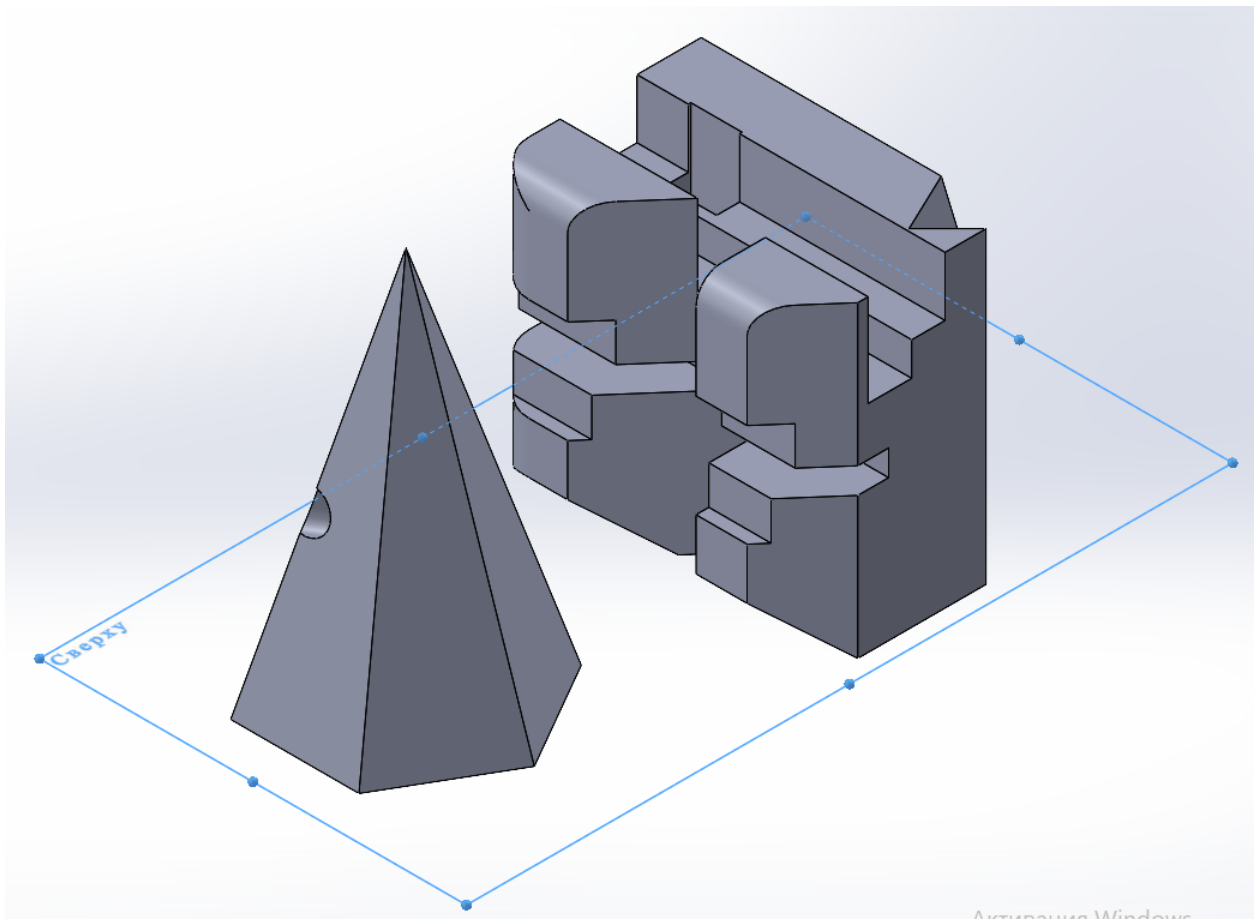


Рисунок 5

Autodesk Windows