

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»

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

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:

Функция	Параметры	Описание	Пример вызова функции
add_item	название: str идентификационный номер: int количество: int	Добавляет товар на склад. Если товар с таким идентификационным номером уже существует, увеличивает его количество.	add_item("Телефон", 101, 10)
remove_item	идентификационный номер: int	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: str	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()
```

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.

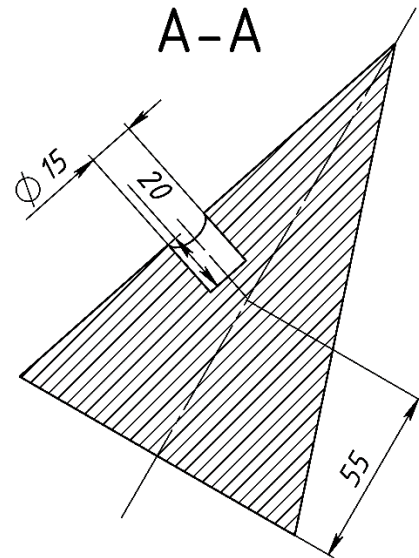
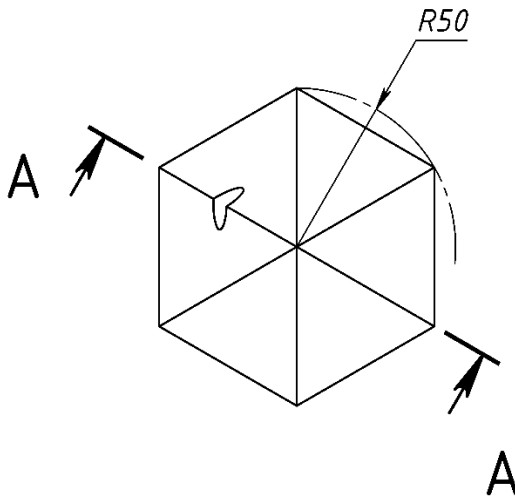
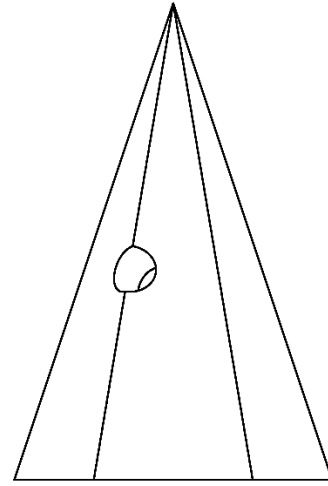
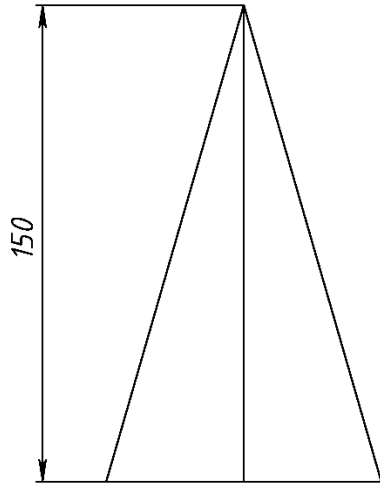
Задание 3.

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

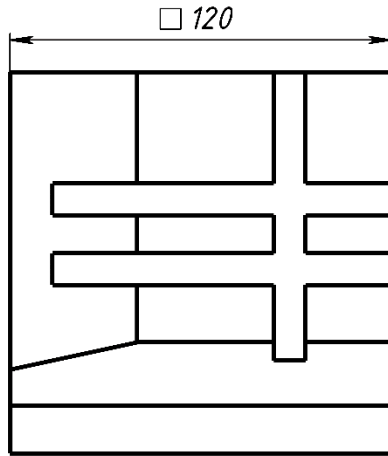
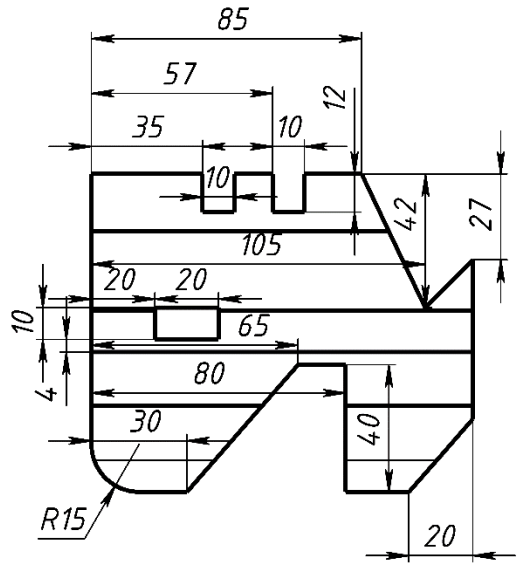
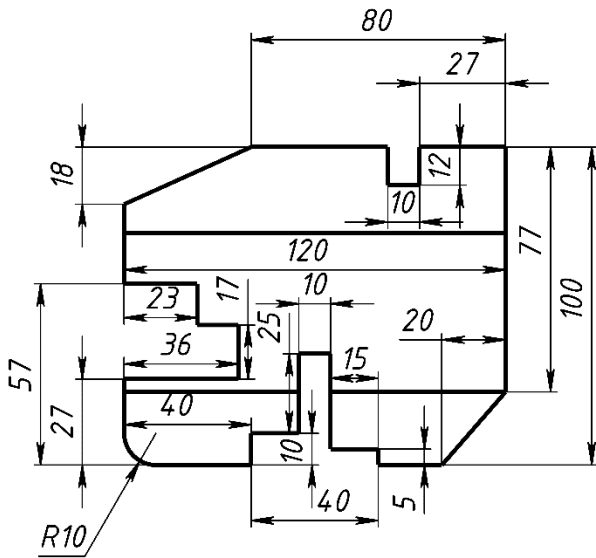
Лист	Масса	Масштаб
		1:2
Лист	Листов 1	
МАИ НИУ		

Копировал

Формат А4

Рисунок 1

Демовариант



Отверстие сквозное

Демовариант

Деталь

МАИ НИУ

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.				

Лит.	Масса	Масштаб
		1:2
Лист	Листов 1	

Копировал

Формат А4

Критерии снижения оценки выполненных заданий

Задание	Критерий	Количество снижаемых баллов
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:

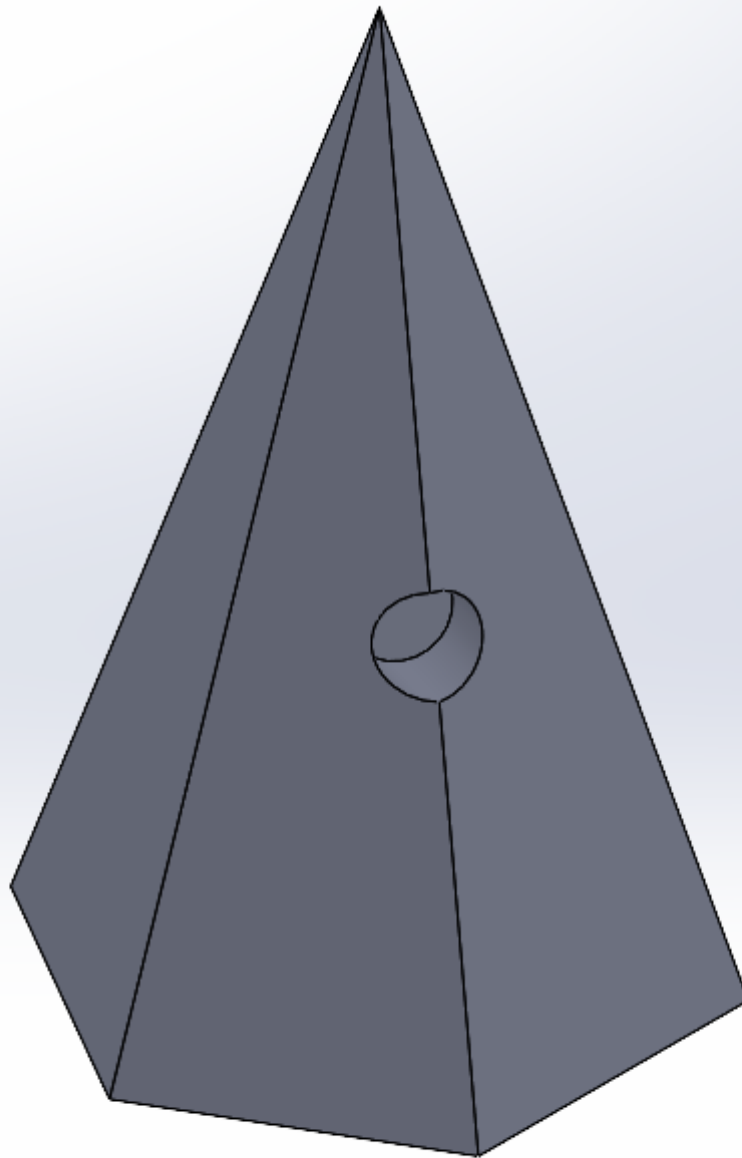


Рисунок 3 - Пример построения примитива

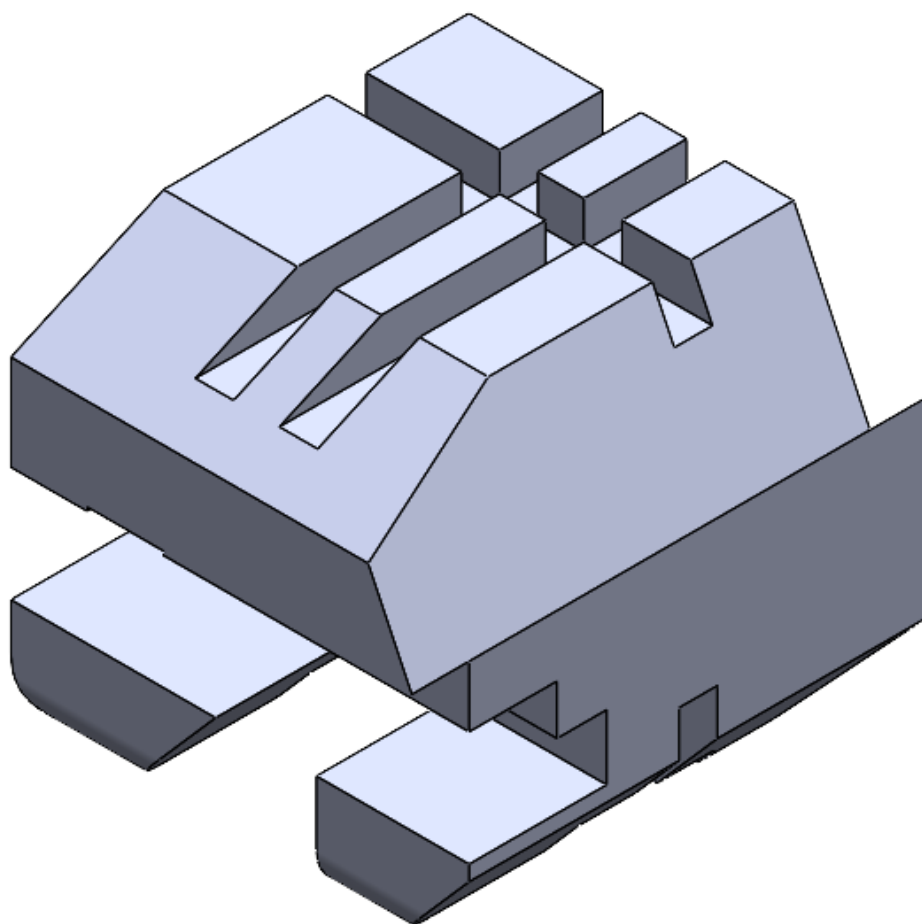


Рисунок 4 - Пример построения детали