Задание 4

ХУАН ЦЗИНЬЯНЬ

4.1 Create a new DataFrame that deletes all rows prior to January 1, 1950 in it.

4.1.py

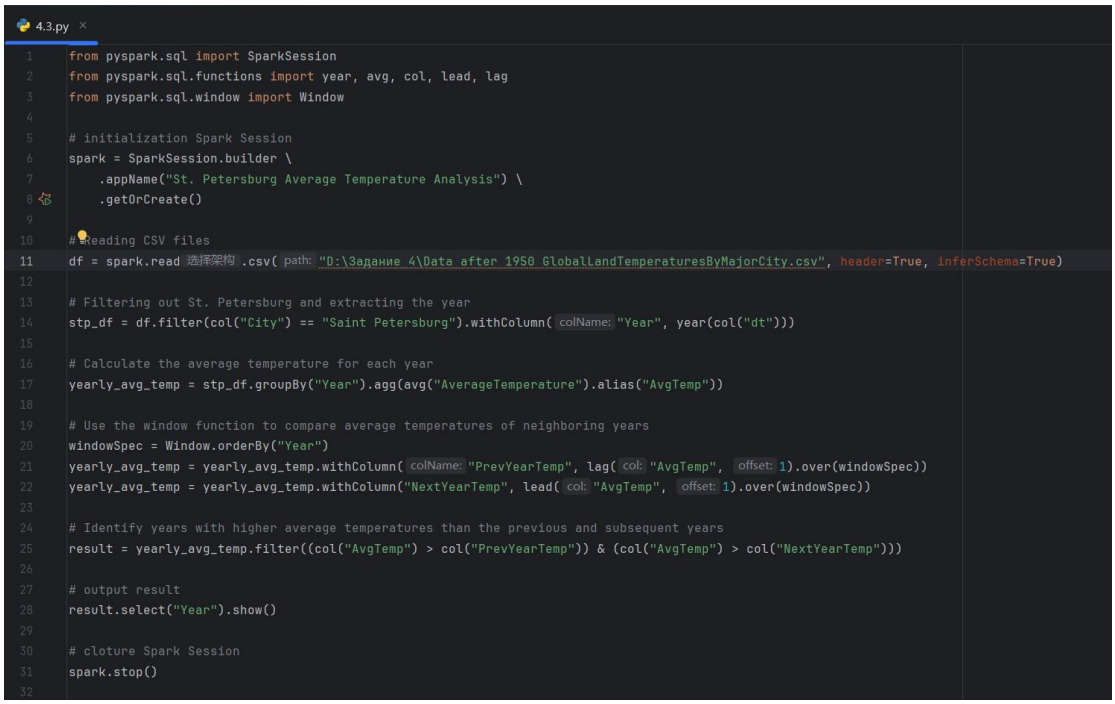
```python
from pyspark.sql import SparkSession
from pyspark.sql.functions import col
import pyspark.sql.functions as F

# initialization Spark Session
spark = SparkSession.builder \
    .appName("Temperature Data Processing") \
    .getOrCreate()

# Reading CSV files
df = spark.read 选择架构 .csv( path: "D:\3адание 4\GlobalLandTemperaturesByMajorCity.csv", header=True, inferSchema=True)

# Show raw data structure
df.printSchema()

# Filter out data after 1950-01-01
filtered_df = df.filter(col("dt") >= "1950-01-01")

# View Results
filtered_df.show()

# If desired, you can save a new DataFrame
# filtered_df.write.csv("FilteredTemperatureData.csv")

# cloture Spark Session
spark.stop()
```

4.2 Find the city with the highest variance of temperature samples in the given data.

4.2.py

```python
from pyspark.sql import SparkSession
from pyspark.sql.functions import var_samp

# initialization Spark Session
spark = SparkSession.builder \
    .appName("Max Variance City in Temperature Data") \
    .getOrCreate()

# Reading CSV files
df = spark.read 选择架构 .csv( path: "D:\3адание 4\Data after 1950 GlobalLandTemperaturesByMajorCity.csv", header=True, inferSchema=True)

# Calculate the sample variance of temperature for each city
variance_df = df.groupBy("City").agg(var_samp("AverageTemperature").alias("Variance"))

# Find the city with the highest sample variance
max_variance_city = variance_df.orderBy( *cols: "Variance", ascending=False).first()

# Print results
print(f"The city with the maximum temperature variance is {max_variance_city['City']} with a variance of {max_variance_city['Variance']}.")

# cloture Spark Session
spark.stop()
```

4.3 Calculate the average annual temperature data for St. Petersburg. Find the years in which the average temperature was higher than the previous and the following year.



4.4 Find out which cities:

1. the difference between the maximum and minimum mean annual temperatures in the sample is the greatest.

2. has the greatest difference between the mean temperature in January and the mean temperature in July.

3. has the greatest number of months with negative mean annual temperatures.

4.4.py

```python
from pyspark.sql import SparkSession
from pyspark.sql.functions import col, year, month, avg, abs, to_date

# initialization Spark Session
spark = SparkSession.builder \
    .appName("City Temperature Analysis") \
    .getOrCreate()

# Reading CSV files
df = spark.read 选择架构 .csv( path: "D:\Задание 4\Data after 1950 GlobalLandTemperaturesByMajorCity.csv", header=True, inferSchema=True)

# Convert a date string to a date type and extract the year and month
df = df.withColumn( colName: "Date", to_date(col("dt"), format: "yyyy-MM-dd")) \
       .withColumn( colName: "Year", year(col("Date"))) \
       .withColumn( colName: "Month", month(col("Date")))

# Registering a DataFrame as a SQL Queryable View
df.createOrReplaceTempView("temperature_data")

# SQL Query 1: Cities with the largest difference between the maximum and minimum annual average temperature
query1 = """
SELECT City, MAX(YearlyAvgTemp) - MIN(YearlyAvgTemp) AS MaxDiff
FROM (
    SELECT City, Year, AVG(AverageTemperature) AS YearlyAvgTemp
    FROM temperature_data
    GROUP BY City, Year
)
GROUP BY City
ORDER BY MaxDiff DESC
LIMIT 1;
"""

# SQL Query 2: Cities with the largest difference between the mean temperature in January and the mean temperature in July
query2 = """
SELECT City, ABS(JanTemp - JulTemp) AS MaxDifference
FROM (
    SELECT City,
           AVG(CASE WHEN Month = 1 THEN AverageTemperature ELSE NULL END) AS JanTemp,
           AVG(CASE WHEN Month = 7 THEN AverageTemperature ELSE NULL END) AS JulTemp
    FROM temperature_data
    GROUP BY City
)
ORDER BY MaxDifference DESC
LIMIT 1;
"""

# SQL Query 3: Cities with the highest number of months with negative average annual temperatures
query3 = """
SELECT City, COUNT(*) AS NegativeMonths
FROM (
    SELECT City, Month, AVG(AverageTemperature) AS MonthlyAvgTemp
    FROM temperature_data
    GROUP BY City, Month
    HAVING MonthlyAvgTemp < 0
)
GROUP BY City
ORDER BY NegativeMonths DESC
LIMIT 1;
"""

# Execute a SQL query and print the results
result1 = spark.sql(query1)
result2 = spark.sql(query2)
result3 = spark.sql(query3)

print("The city with the largest difference between the maximum and minimum annual average temperature of the city:")
result1.show()

print("Cities with the largest difference between the average temperature in January and the average temperature in July:")
result2.show()

print("Cities with the highest number of months with negative average temperatures throughout the year:")
result3.show()

# cloture Spark Session
spark.stop()
```