Maryam Soleimani

PI:
I used the Chudnovsky algorithm to calculate pi.
PI class: we have BigDecimal d and an int floatingPoint, that shows the number of digits after the decimal point pi.
run method: calculates a term of the Chudnovsky algorithm using the factorial method and updates the answer by using the sumInc method.
The factorial method calculates the factorial of a given int using the BigDecimal class.
The sumInc method updates the answer by incrementing a given BigDecimal to it.
The calculate method: takes an int floatingPoint that shows the number of digits after the decimal point for the calculated value of pi. It creates a thread pool using the Executors.newCachedThreadPool() method and submits PI objects to the thread pool to calculate.

Priority Simulator:
Black, Blue, and WhiteThread each of them have count down latch that are used to synchronize the threads and make sure that they execute in an order.
in the run method after printing the message, the thread calls the countdown method.
The Runner class creates a list of ColorThread objects and initializes three CountDownLatch objects for each thread type. The Runner class then creates an instance of each thread type and adds it to the list of threads. The threads are started, and the CountDownLatch objects are awaited to make sure that the threads execute in the right order.

Semaphore:
The Operator class:It takes a Semaphore object and a name as args. The Semaphore object is used to restrict access to the critical section to a maximum of two operators at a time. I called acquire method on the Semaphore object to acquire the permit, and the release method to release the permit after the critical section is executed.
The Resource class: accessResource() takes a name as an argument and simulates accessing a shared resource by printing the name and the current time.
The Controller class: in the main()a Semaphore object with a permit count of 2 is created and it is set to true
The Semaphore object ensures that a maximum of two operators can access the shared resource at the same time, which prevents race conditions.