

Zadanie 1.

1. Zdefiniować klasę `MyThread1` jako specjalizację klasy `Thread`.
2. Zdefiniować konstruktor z parametrem `String` jako nazwa obiektu.
3. W metodzie `run()` wypisać 10 razy komunikat: „Pozdrowienia z wątku ” + `getName()`.
4. Zdefiniować klasę `Test1` z metodą `static void main(String[] args)`, w której należy powołać:
5. Tablicę 10 elementową obiektów typu `Thread`.
6. Powołać obiekty tablicy jako wątki typu `MyThread1`.
7. Uruchomić wątki.
8. Zaczekać na zakończenie wątków potomnych wywołując metodę `join()`.
9. Wypisać komunikat „Koniec”.

Zadanie 2.

1. Zdefiniować klasę `MyThread2` jako implementującą interfejs `Runnable`.
2. W metodzie `run()` wypisać 10 razy komunikat: „Pozdrowienia z wątku ” + `getName()`.
3. Zdefiniować klasę `Test2` z metodą `static void main(String[] args)`, w której należy powołać:
4. Tablicę 10 elementową obiektów typu `Thread`.
5. Powołać obiekty tablicy jako wątki typu `Thread` opakowujące obiekty typu `MyThread2` (przekazać je przez konstruktor klasy `Thread`).
6. Uruchomić wątki.
7. Zaczekać na zakończenie wątków potomnych wywołując metodę `join()`.
8. Wypisać komunikat „Koniec”.

Zadanie 3.

1. Zdefiniować klasę `MyThread3` jako specjalizację klasy `Thread`.
2. Zdefiniować konstruktor z parametrem `String` jako nazwa obiektu.
3. W metodzie `run()` w pętli `for` 10-krotnie:
 - a. w bloku `try` za pomocą metody `sleep` uśpić wątek na 1s i po obudzeniu wypisać komunikat: „Spałem przez 1s - ” + `getName()`.
 - b. w ramach obsługi wyjątku `InterruptedException`:
 - i. wypisać komunikat: „Zostałem obudzony - ” + `getName()`.
 - ii. Zakończyć działanie metody `run()`.
4. Zdefiniować klasę `MyThread4` jako specjalizację klasy `Thread`.
5. Zdefiniować konstruktor z parametrem `String` jako nazwa obiektu.
6. W metodzie `run()`
 - a. Powołać i uruchomić wątek potomny klasy `MyThread3`.
 - b. w pętli `for` 5-krotnie:
 - c. w bloku `try` za pomocą metody `sleep` uśpić wątek na 2s i po obudzeniu wypisać komunikat: „Spałem przez 2s - ” + `getName()`.
 - d. w ramach obsługi wyjątku `InterruptedException`:
 - i. wypisać komunikat: „Zostałem obudzony - ” + `getName()`.
 - ii. wywołać metodę `interrupt` drugiego wątku potomnego.
 - iii. Zaczekać na zakończenie wątku potomnego wywołując metodę `join()`.
 - iv. Zakończyć działanie metody `run()`.
7. Zdefiniować klasę `Test3` z metodą `static void main(String[] args)`, w której należy powołać:
8. Powołać obiekt wątku typu `MyThread4`.
9. Uruchomić wątek.
10. w bloku `try` za pomocą metody `sleep` uśpić wątek główny na 4,5s i po obudzeniu wywołać metodę `interrupt` wątku potomnego.
11. Zaczekać na zakończenie wątku potomnego wywołując metodę `join()`.
12. Wypisać komunikat „Koniec”.

Zadanie 4.

1. Zdefiniować klasę `Licznik` z prywatnym atrybutem typu `long`, publicznymi metodami: `get()` oraz `inc()`.
2. Zdefiniować klasę `MyThread5` jako specjalizację klasy `Thread`.
3. Zdefiniować konstruktor klasy `MyThread5` z parametrem `String` jako nazwa obiektu oraz konstruktor klasy `Licznik`.
4. W metodzie `run()` w pętli `for` 5000000 razy wywołać metodę `inc()` na obiekcie współdzielonym licznik.
5. Zdefiniować klasę `Test4` z metodą `static void main(String[] args)`, w której należy powołać:
6. Tablicę 10 elementową obiektów typu `Thread`.
7. Powołać obiekty tablicy jako wątki typu `MyThread4`.
8. Uruchomić wątki.
9. Zaczekać na zakończenie wątków potomnych wywołując metodę `join()`.
10. Wypisać komunikat „Stan licznika = ” + `licznik.get()`.