

Letter Box

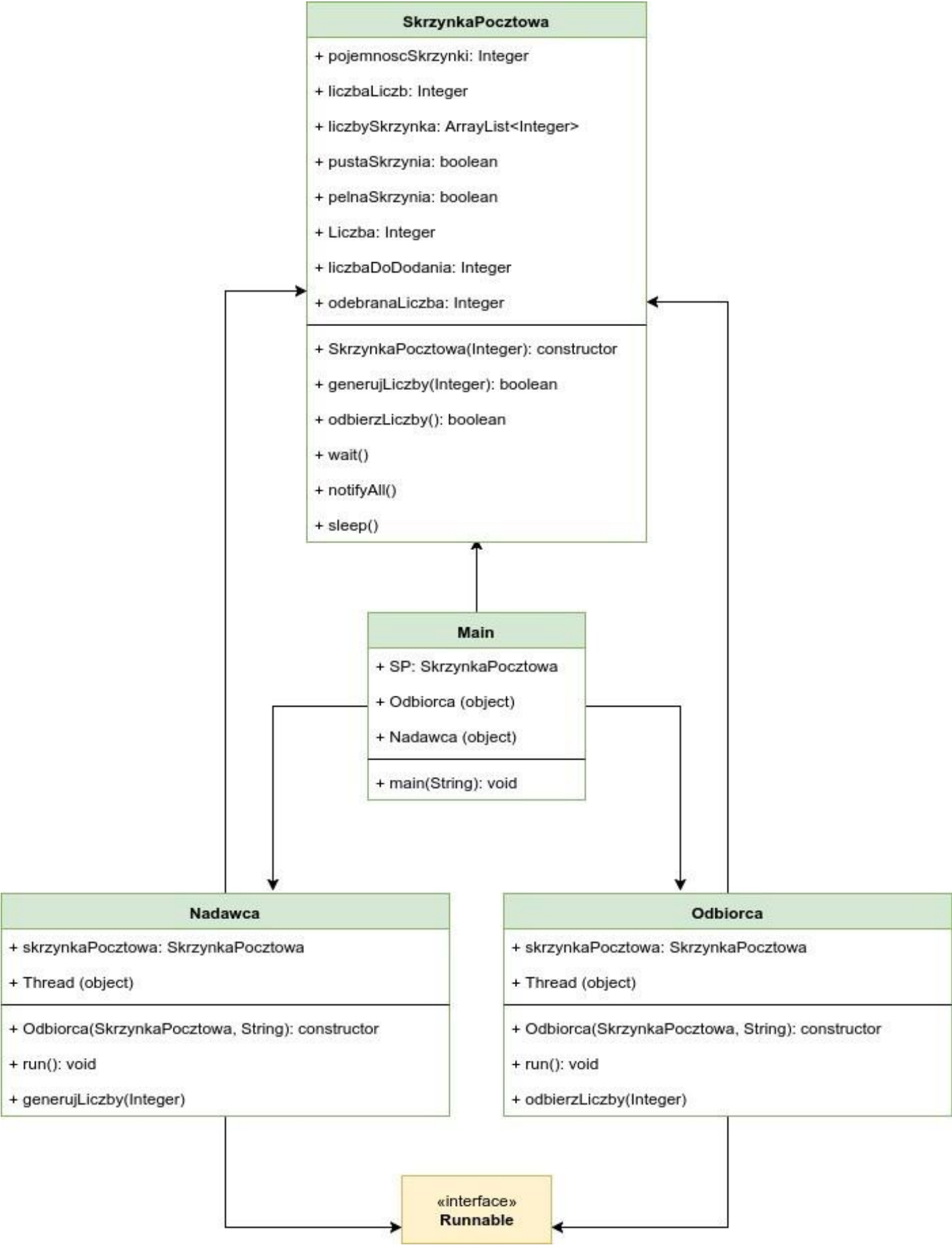
documentation

Daniel Wrona

TOPIC

Write a concurrent program consisting of five processes. The SP process acts as a mailbox with a capacity of n integers. Processes P1 and P2 generate integers and send them one by one to SP. The P3 and P4 processes take one number from the box. The value of n is the program parameter.

UML

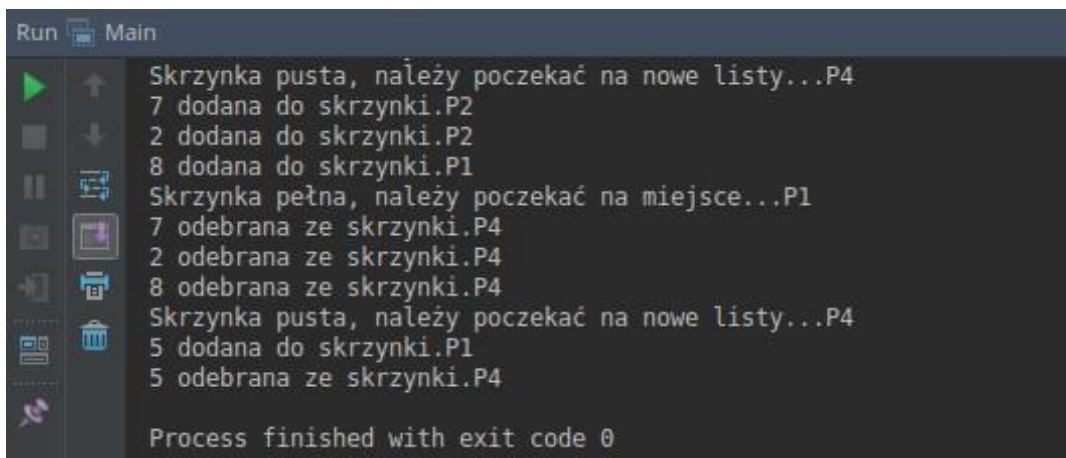


Classes

- **SkrzynkaPocztowa** – main program logic. It contains the 'generujLiczby' and 'odbierzLiczby' methods that describe the operation of the entire program.
- **Nadawca** – implements the Runnable interface and describes the behavior of p1 and p2 processes that generate numbers to the mailbox using the logic of the 'generujLiczby' method.
- **Odbiorca** – implements the interface Runnable and describing p3 and p4 processes that receive data using the methodology of the 'odbierzLiczby' method.
- **Main** – the main class of the program that runs it. It creates five processes, one acts as a mailbox, two generators and two recipients of numbers.

How it working

The main logic of the program was included in the class SkrzynkaPocztowa. Mechanisms for adding and receiving numbers from the mailbox have been implemented. Checks if the mailbox is empty - if so, the process of receiving numbers is stopped by the "wait" method and when the numbers appear in the box, it is resumed by the "notifyAll" method. Similarly, it checks whether the box is not full - if so, the process of generating numbers is stopped by the method „wait” and when the space for subsequent lists appears in the mailbox, is resumed by „notifyAll” method. The capacity of the box determines the parameter that is passed in the constructor. The box mechanism was built based on a list to which the numbers generated are added or received in sequence. Main class runs the entire program, creating 5 processes. Nadawca i Odbiorca classes allow you to create new processes to generate and receive numbers from the mailbox. The method "sleep" was also used, which allows the thread to perform the operation from time to time (in the program 1 second).



```
Run Main
Skrzynka pusta, należy poczekać na nowe listy...P4
7 dodana do skrzynki.P2
2 dodana do skrzynki.P2
8 dodana do skrzynki.P1
Skrzynka pełna, należy poczekać na miejsce...P1
7 odebrana ze skrzynki.P4
2 odebrana ze skrzynki.P4
8 odebrana ze skrzynki.P4
Skrzynka pusta, należy poczekać na nowe listy...P4
5 dodana do skrzynki.P1
5 odebrana ze skrzynki.P4
Process finished with exit code 0
```

A screenshot from the example run of the program