

Interfejs `Iterable` to innymi słowy pętla iteracyjna, w innych językach programowania znana jako pętla *foreach*. Umożliwia przeglądanie wszystkich elementów należących do danej kolekcji (np. do biblioteki użytej w laboratorium 2). Pętla może być stosowana z tablicami oraz klasami, które implementują interfejs `Iterable <T>`. Obiekt klasy `Iterator<T>` umożliwia wykorzystanie metod, np. `hasNext()`, `next()`. Przykładowy fragment kodu, który miałem wykorzystać w projekcie drugim, ale postawiłem na prostszą metodę:

```
Iterator<Ksiazka> i = BibliotekaZasobow.iterator();
while(i.hasNext()){
    Ksiazka k = (Ksiazka) i.next();
    total = total + b.toString();
}
return total;
```

Interfejs `closeable` to interfejs wykorzystywany do zwalniania zasobów w sytuacji, gdy w programie wystąpi wyjątek. Interfejs `closable` może wyrzucać tylko wyjątek `IOException`. Interfejs `Closeable` jest przestarzały, został zastąpiony przez interfejs `AutoCloseable`. Ogólnie, służy do zwalniania zasobów, których używa dany obiekt (np. otwarte pliki). Przykładowy kod:

```
void close()
    throws IOException
```

zamyka np. pętlę i zwalnia wszystkie zasoby z nią powiązane. Jeśli np. pętla jest już zamknięta, ta metoda nic nie robi.