PWr Spotkanie 7
Aplikacje webowe na platformie .NET
Laboratorium – **Lista 7**

Wstęp.

Język C# posiada mechanizm wyjątków, które rzuca się poprzez instrukcję **`throw`**, natomiast łapie się w instrukcji **`try/catch/finally`**. Posiada również instrukcję:

```
using(resource) { <instructions> }
```

którą kompilator przerobi w języku pośrednim na kod podobnego do poniższego:

```
{
  resource;
  try{
     <instruction>
  }
  finally{
    resource?.Dispose();
  }
}
```

Oznacza to, że zasób `resource` musi posiadać metodę `void Dispose()`, a konkretnie musi implementować interfejs `IDisposable` z tylko tą jedną metodą. Jednak w przypadku tworzenia własnych klas implementujących ten interfejs należy stworzyć jeszcze kilka innych metod i stosować się do schematu opisywanego na stronie:
https://docs.microsoft.com/pl-pl/dotnet/api/system.idisposable?view=netcore-3.1
Wymagane jest to szczególnie dla zasobów niezarządzanych przez maszynę wirtualną .Net. Do tego typu zasobów należą pliki, klasy dostępu do bazy danych, drukarki itd.

Pamiętać należy, że jeśli użyjemy `using` wraz z tworzeniem zmiennej np.:

```
using(var file=…) { <instructions> }
```

To zakres zmiennej **`var`** ograniczy się tylko do użycia w bloku `{ <instructions> }`.
Jeśli jednak zostanie zadeklarowana wcześniej np.:

```
var file=…;
using(file) { <instructions> }
...
```

to po wykonaniu bloku instrukcji nadal mamy dostęp do tej zmiennej, ale albo jej wartość wynosi **`null`** albo zasób został i tak zwolniony (co najczęściej oznacza jego bezużyteczność). Ten sposób korzystania z zasobu uznaje się za złą praktykę.
Od języka C# 8.0 można używać skróconego zapisu:

```
using var variable=…;
<instructions>
...
```

W tym przypadku wywołanie metody `Dispose()` nastąpi po wyjściu z bloku instrukcji o zasięgu zmiennej `variable`.

List zadań

Przygotować program demonstrujący użycie zaimplementowanych rozwiązań dla różnych przypadków użycia.
Wykorzystać kolekcje i operacje na nich obecne w standardowej bibliotece C#.

1. Napisać program, który wczytuje z klawiatury ścieżkę do pliku tekstowego. Następnie odczytuje plik i rozbija go na słowa. Za słowa uznajemy niezerowy, nieprzerwany ciąg znaków z zakresu od ‘a’ do ‘z’ i od ‘A’ do ‘Z’ (tylko z alfabetu angielskiego) W wyrazach wszystkie znaki liter mają zostać zamienione na małe litery. Stworzyć słownik zliczający ile razy w pliku wystąpił dany wyraz. Posortować wyrazy wg częstości występowania. Wypisać 10 (lub mniej jeśli nie było 10 różnych słów) słów od najczęściej występujących wraz z liczbą wystąpień. Dla testów skopiować sobie pliki z adresów:
   http://norvig.com/big.txt
   http://www.gutenberg.org/
2. W przypadku podania nieprawidłowej nazwy pliku za pomocą mechanizmu wyjątków wyłapać tą sytuację. Jakie jeszcze wyjątki mogą się pojawić podczas operacji na pliku, dopisać kod obsługi takiej sytuacji, postarać się sprowokować takie błędy.
3. Czy część z wyjątków z zadania 2 można „wyłapać” bez uruchamiania mechanizmu wyjątków? Jeśli tak stwórz drugą wersję tego zadania, bez tych wyjątków.

Data I: Spotkanie 8 (max 100 punktów)
Data II: Spotkanie 9 (max 80 punktów)
Data III: Spotkanie 10 (max 50 punktów)