

15. Wyjątki

1 Wyjątki

Wyjątki (ang. *exceptions*) opisują pewną nieprawidłową, błędną lub wyjątkową sytuację. Wiążą się z nimi dwie czynności:

- wyrzucanie (inaczej zgłoszenie) wyjątku (ang. *throw*),
- obsłużenie (inaczej złapanie) wyjątku (ang. *catch*).

Wyjątki w języku Java są obiektami klasy `Throwable` z pakietu `java.lang` lub jej dowolnej podklasy (najczęściej `Exception`). Wyróżniamy dwa typy wyjątków: niekontrolowane i kontrolowane.

Zadanie 01 Zapoznaj się z kodem zawartym w pliku `StackAndExceptions.java`. Skompiluj go, a następnie uruchom bez argumentów program i z kilkoma argumentami:

```
1 java StackAndExceptions one two three four five
```

Jaki i jakiego typu wyjątek zostaje zgłoszony gdy program jest uruchamiany z więcej niż trzema argumentami? Jak do tej pory obsługiwałeś tego typu wyjątkowe sytuacje?

1.1 Rzucanie wyjątków

Rzucenie obiektu wyjątku odbywa się poprzez słowo kluczowe `throw`:

```
1 if (array.length > 3)
2     throw new RuntimeException("Too many elements.");
```

Zadanie 02 Zmodyfikuj metodę `push()` klasy `Stack` w taki sposób, aby rzucała wyjątek klasy `RuntimeException`, w sytuacji gdy nie można dodać już elementu do stosu. Czy poza zmianą komunikatu błędu zgłaszanego przez wirtualną maszynę Javy, działanie programu zmieniło się?

1.2 Obsługa wyjątków

Za obsługę wyjątków odpowiada klauzula `try-catch`:

```
1 try {
2     // kod, w którym może zostać rzucony wyjątek
3 }
4 catch (Exception e) {
5     // kod wykonany, gdy zostanie zgłoszony wyjątek
6 }
```

Po słowie kluczowym `catch` następuje deklaracja zmiennej dla obiektu wyjątku, z której będzie można skorzystać wewnątrz bloku `catch`. Jest to klasa wyjątku, który chcemy obsłużyć lub jego klasa nadrzędna.

Zadanie 03 Obsłuż wyjątek `RuntimeException` w metodzie `main()` w taki sposób, aby wypisać na ekran pożądaný komunikat bez przerywania całego programu.

1.3 Propagacja wyjątków

Podczas wystąpienia wyjątku w metodzie *A*, jej wykonanie jest przerywane i sterowanie powraca do metody, w której metoda *A* została wywołana. Jeżeli to miejsce kodu znajduje się w bloku `try`, wyjątek jest obsługiwany, w przeciwnym przypadku wyjątek propagowany jest dalej, aż do obsłużenia przez wirtualną maszynę Javy.

Każda metoda, w której może wystąpić wyjątek kontrolowany, a która sama go nie obsługuje, musi zawierać o tym informacje po słowie kluczowym `throws`, np.:

```
1 void foo() throws Exception {  
2     if (somethingBadHasHappened())  
3         throw new Exception("Something bad has happened!");  
4 }
```

Zadanie 04 Zamień klasę wyjątku rzucanego w metodzie `push()` z `RuntimeException` na `Exception`. Czy teraz można skompilować kod programu?

Wyjątek `Exception` to wyjątek kontrolowany, więc dodaj do metody `push()` klauzulę `throws`.

Zadanie 05 Dopisz klasy `FullStackException` oraz `EmptyStackException`, które będą dziedziczyły po klasie `Exception` i przesłaniały (redefiniowały) metodę `getMessage()`, która zwraca komunikat wyjątku.

Zadanie 06 Zmodyfikuj metody `push()` oraz `pop()` tak, aby w sytuacjach wyjątkowych zgłaszały wcześniej utworzone wyjątki. Dodaj metodę `swap()`, która zamieni miejscami dwa górne elementy stosu. W jej implementacji wykorzystaj poprzednie metody.

Użyj jej w metodzie `main()`, pamiętaj o dodaniu klauzuli `throws` w odpowiednich miejscach.

Zadanie 07* Zmodyfikuj program tak, aby elementy do stosu były wczytywane z pliku (jedna linia — jeden element) o nazwie podawanej jako parametr programu. Klasa `Stack` powinna posiadać metodę `readElementsFromFile(String fileName)`, która zostanie wykorzystana w konstruktorze jednoargumentowym.

Wyjątki związane z błędnym czytaniem pliku powinny być obsługiwane w metodzie `main()`.

Zadanie 08 dodatkowe (1 pkt) Zaimplementuj powyższy program z klasą `Stack` w języku C++ (bez zadania 07.), definiując i obsługując własne klasy wyjątków.