

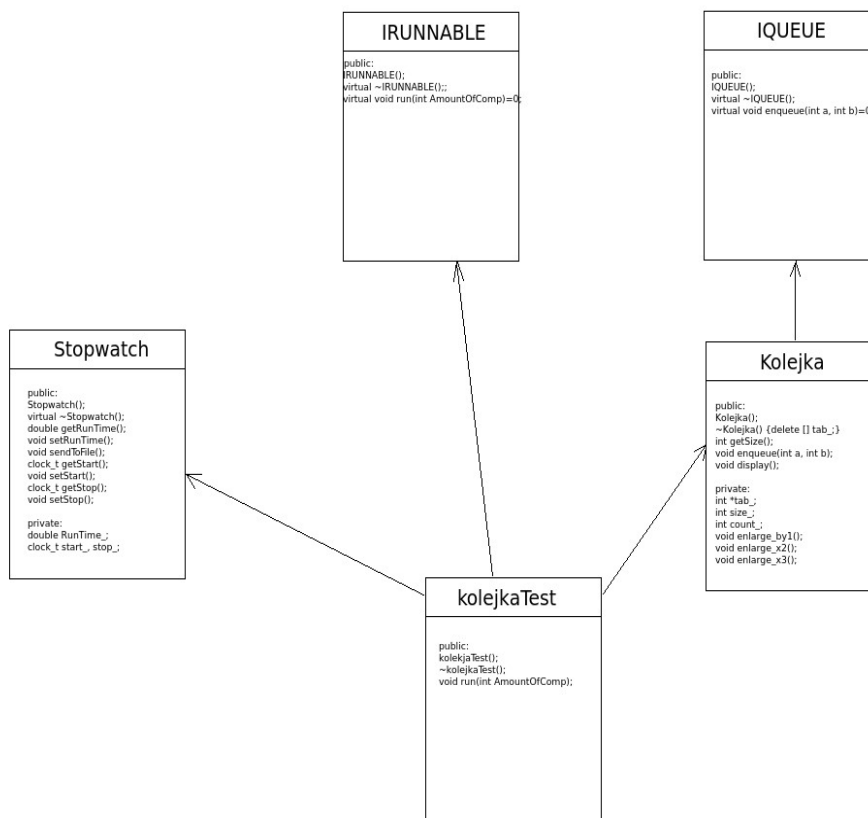
Projektowanie algorytmów i metody sztucznej inteligencji

prowadzący:
Mgr inż. Andrzej Wytyczak – Partyka

1. Zadanie do zrealizowania:

Należało zmodyfikować poprzedni program w taki sposób, aby działał zgodnie ze wzorcem Mostu. Wzorec ten oddziela abstrakcję od jej implementacji. Pozwala to na usunięcie powiązań pomiędzy implementacją, a obiektami, które ją wykorzystują, co powoduje, że abstrakcja i implementacja mogą zmieniać się niezależnie od siebie. W tym celu należało stworzyć interfejs Iqueue oraz Irunnable, klasę Stopwatch w celu dokonywania pomiarów.

2. Diagram klas:



3. Uzyskane wyniki:

strategia zwiększania o 1:

Liczba elementów	Czas [sek]
10	0,00000218
10^3	0,00213718
10^5	17,91061297
10^6	1 787,50092108
10^9	brak*

*po ponad 7 godzinach wciąż nie otrzymano wyników



Strategia zwiększania x2:

Liczba elementów	Czas [sek]
10	0,00000173
10^3	0,00002145
10^5	0,00156427
10^6	0,01508000
10^9	17,08789141



Strategia zwiększania x3:

Liczba elementów	Czas [sek]
10	0,00000118
10 ³	0,00001636
10 ⁵	0,00171509
10 ⁶	0,01457036
10 ⁸	1,46015601



Zestawienie trzech strategii:



4. Wnioski:

Wzorzec Mostu jest bardzo przydatny w programowaniu obiektowym, pozwala na:

- Ukrycie implementacji przed klientem. Metody abstrakcji wywołują metody dostępne przez interfejs implementacji. Tworząc nową klasę pochodną Abstrakcji nie odwołujemy się bezpośrednio do implementacji, a jedynie do metod klasy bazowej abstrakcji, które stanowią opakowanie dla metod implementacji;
- Łatwe rozbudowywanie abstrakcji i implementacji poprzez dodanie nowych klas pochodnych;
- Niepowiązywanie abstrakcji na stałe z konkretną implementacją, wiązanie jest dokonywane w momencie tworzenia nowego obiektu, może być również zmieniane w po utworzeniu obiektu;
- Współdzielenie implementacji przez wiele obiektów. Abstrakcja przechowuje wskazanie na klasę bazową implementacji, co umożliwia wykorzystanie jednej implementacji przez wiele obiektów.