**Wojskowa Akademia Techniczna**

**Systemy Baz Danych**

**Sprawozdanie**

**Obiektowe bazy danych**

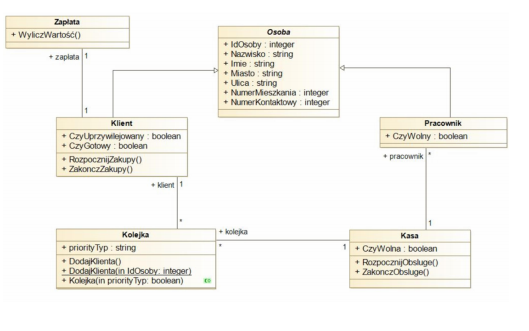
Wykonał: Jankowski Łukasz, Leśniewski Konrad

Grupa: I8E1S4

Dla wybranego zagadnienia należy :

1. Identyfikacja klas, metod, związków między klasami dla wybranego obszaru dziedzinowego
2. Implementacja obiektowej bazy danych dla wybranego obszaru dziedzinowego wykorzystując Cache Intersystems:
   * model bazy danych powinien składać się z co najmniej 6 klas;
   * model bazy danych powinien zawierać:
     + metody;
     + dziedziczenie;
     + związki;
     + klasy abstrakcyjne i polimorfizm;
     + nowo zdefiniowane typy danych;
3. Wypełnienie bazy danych testowymi danymi
4. Pokaz wykonania metod obiektów w terminalu
5. Umieszczenie projektu w repozytorium git np. na <https://github.com/>. Wymagana jest aktywność wszystkich uczestników grupy.
6. Udokumentowanie pracy w formie sprawozdania laboratoryjnego oraz filmiku z działania aplikacji np. na youtube. Długość filmiku do ~ 7 minut
7. Podsumowanie wykonanej pracy w szczególności uwzględniając:
   * napotkane problemy i ich rozwiązania,
   * pomysły/idee,
   * ocenę środowiska.
   * wypełnioną tabelkę zamieszczoną na końcu tego dokumentu

**Tematem projektu będzie obsługa sklepu**

**Model bazy danych**

Baza wykorzystuje klasy:

• Osoba – klasa abstrakcyjna. Zawiera dane osobowe dotyczące danego obiektu.

• Klient – klasa dziedzicząca z klasy Osoba, dodane atrybuty:

*CzyUprzywilejowany* – charakteryzuje obiekt, który ma pierwszeństwo w kolejce

*CzyGotowy* – charakteryzuje obiekt czy wykonał już przypisane mu operacje *RozpocznijZakupy()* oraz *ZakonczZakupy()*

*RozpocznijZakupy()* - metoda rozpoczynająca proces zakupów, losuje czas trwania zakupów w zakresie od 0 do 5 sekund

*ZakonczZakupy()* - kończy proces zakupów i ustawia parametr klienta *CzyGotowy* na TRUE, metoda odczekująca wylosowany czas, symulując trwanie zakupów

• Pracownik – klasa dziedzicząca z klasy Osoba, rozszerza ją przez dodanie atrybutu *CzyWolny*, określający czy dany pracownik może iść obsługiwać klientów do kasy

• Kasa – klasa definiująca obiekty określające miejsce pracy pracownika w celu obsługi kolejki klientów, posiada ono atrybuty:

*CzyWolny* - parametr odpowiadający czy dana kasa jest wolna i czy kolejny klient z kolejki może być obsługiwany

*RozpocznijObsługe()* - metoda rozpoczynająca proces obsługi, losuje czas trwania obsługi w zakresie od 0 do 5 sekund,

*ZakonczOblsuge()* - kończy obsługę i ustawia parametr kasy *czyWolne* na TRUE, metoda odczekująca wylosowany czas, symulując trwanie obsługi

• Kolejka – klasa definiująca element odpowiadający kolejce klientów, charakteryzowany przez atrybut *priorityTyp,* opisujący czy dana kolejka jest priorytetowa i przepuszcza uprzywilejowanych klientów w procesie obsługi oraz wykonuje on metody polimorficzne:

*Dodaj()*

*Dodaj(IdOsoby)*

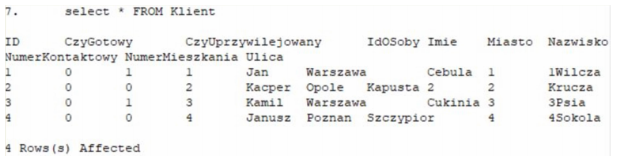
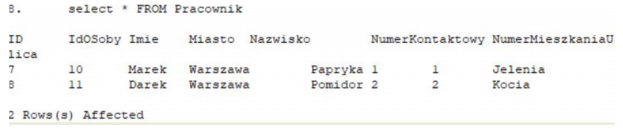
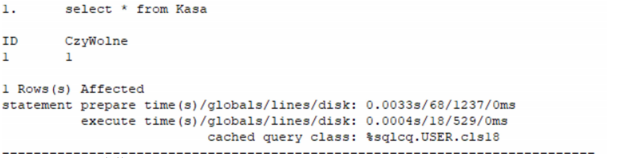
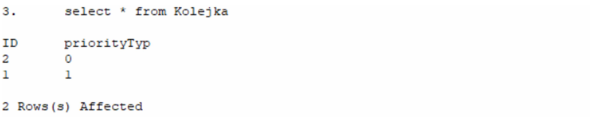
mające za zadanie dodać klienta do kolejki.

• Zapłata – klasa definiująca element odpowiadający za wyliczenie wartości zakupów klienta. Wyliczeniu wartości odpowiada wylosowanie wartości w zakresie od 0 do 50, dla ułatwienia rozwiązania, poprzez metodę *WyliczWartość().*

**Wypełnienie bazy danych**

Dla każdej klasy stworzyliśmy pojedyncze obiekty, aby tylko przedstawić poprawność polecenia oraz później pokazać wykonywanie pojedynczych metod. Proces ten został wykonany dla każdego przedstawionego poniżej elementu.

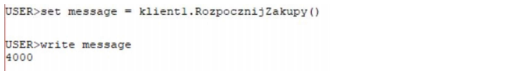
Efekty wypełnienia klas:

* **klient**
* **pracownik**
* **kasa**
* **kolejka**

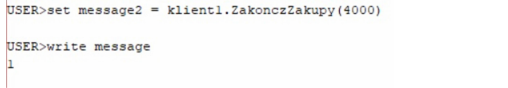
Dla klas Zapłata oraz Osoba nie przedstawiliśmy powyżej wywołań stworzonych obiektów. Jest to spowodowane tym, że klasa Osoba jest klasą abstrakcyjna, jedynie służącą do przechowania definicji Osoby, po której dziedziczą klasy Klient oraz Pracownik odpowiednie parametry, a klasa Zapłata jest jedynie klasą operacyjną posiadająca metodę, więc jej powołane obiekty posiadają tylko ID przypisywane automatycznie przez program.

**Wykonywanie metod:**

W celu sprawdzenia funkcjonalności należy utworzyć odpowiedni obiekt, a następnie wywołać przypisaną do niego metodę. Do pokazania funkcjonowania zaimplementowanych metod użyliśmy przykładów bazujących na:

* *RozpocznijZakupy()* oraz *ZakonczZakupy()* z klasy Klient

Metoda ta wylosowuje czas trwania zakupów z przedziału od 0 do 5 sekund. Wartość w konsoli jest podana w milisekundach. Następnie należy wykonać metodę *ZakonczZakupy()*, podając na wejściu wartość wygenerowaną przez poprzednią metodę, która będzie decydować ile program ma czekać zanim ustawi parametr Klienta *CzyGotowy* na TRUE, aby ten mógłby być dołączony do kolejki. Proces ten ma na celu symulacje wykonywanie zakupów przez klienta.



Obiekty w naszej bazie posiadają także metody, których powyżej nie przedstawiliśmy, są to metody *WyliczWartość(),* *RozpocznijObsluge(),* *ZakonczObsluge()*, znajdujące się w klasach Kasa oraz Zapłata. Ich budowa działania oraz wyniki bazują na tym samym algorytmie, więc wyświetlane wyniki będą takie same.

**Podsumowanie**

Problemy, które napotkaliśmy wynikały głównie z ubogiej wiedzy oraz niskiego doświadczenia z tym typem bazy danych. Praca w tym środowisku jest trudna ze względu na jego małą popularność oraz trudny dostęp do informacji. Jest dokumentacja ale poruszanie się po niej nie należy do rzeczy łatwych.

**Tabela**

*Jak środowisko realizuje zagadnienia związane z:*

* Obiektowością – Metodami

Metody implementuje się w danej klasie. Można wybrać implementacyjny język.

* Obiektowością – Dziedziczeniem, typami abstrakcyjnymi

Dziedziczenie należy zatwierdzić w procesie tworzenia klasy, a następnie utworzyć poprzez użycie parametrów, o typie relacji, odpowiednie połączenia pomiędzy klasami.

* Obiektowością – Związkami między klasami (asocjacja, kompozycja)

Tak jak poprzednik, poprzez utworzenie parametrów, o typie relacji.

* Obiektowością – Polimorfizmem Typowe powielanie metod w klasie, przy zmianie warunków wywołania np. dodatkowego atrybutu wywołania
* Obiektowością – Enkapsulacją

Środowisko umożliwia zarządzanie widocznością części danej klasy poprzez użycie podstawowej metody dopisywania public / private

* Obiektowością – Trwałością danych

Środowisko wykorzystuje stopniowy mechanizm zapisu danych, który opisałem w podpunkcie *„Wypełnienie bazy danych testowymi danymi”*

* Administracja – Zarządzanie środowiskiem

Do projektowania używane jest Cache Studio, a no wykonywaniu zapytań w języki SQL, jak i wykonywaniu zaimplementowanych metod używany jest Cache Terminal

* Interfejs – Czy narzędzie posiada API? Dla jakich języków?

Tak środowisko posiada API. Dla języków java, .NET, perl, python.

* Środowisko – Czy narzędzie zawiera w sobie środowisko programistyczne?

Tak środowisko posiada obszar kompilacyjny, pozwala także na korzystaniu z wielu języków implementacyjnych

* Skalowalność – Czy narzędzie umożliwia horyzontalne skalowanie środowiska? (rozproszone przetwarzanie, magazynowanie, replikacja)

Tak posiada skalowalność. Multi-model – Czy narzędzie zapewnia inne rodzaje bazy danych? Tak posiada multi-model.