

POLITECHNIKA ŚWIĘTOKRZYSKA W KIELCACH			
Wydział: WEAiI	Gr. 312B	Zespół numer 5	25.01.2021
Sprawozdanie z projektu z Podstaw Inżynierii Programowania		Weronika Wolszczak Daniel Ostrowski Szymon Gierasimowicz Piotr Czajkowski	

## Temat projektu: Przedszkole

### Opis systemu:

W ramach projektu stworzyliśmy system zarządzania przedszkolem. Zapewnia on przechowywanie informacji oraz kontrolę nad grafikiem, jadłospisem a także listą pracowników oraz dzieci. Dodatkowo dyrektor ma możliwość przechowywania spisu zabawek a opiekunka dodawania spotkań z rodzicami przypisanymi do poszczególnych dzieci. Rolę „admina” systemu pełni Dyrektor, ma on dostęp do większości klas i metod, może między innymi dodawać i usuwać opiekunki i dzieci z systemu.

### Struktura systemu, opis klas:

W większości przypadków ustawienie i odczytywanie danych odbywa się według metod zaczynających się od słów kluczowych. Dodatkowe funkcje lub odbiegające od przyjętego schematu są wypisane w każdej z klas. Schemat metod pobierających i ustawiających danych został przedstawiony poniżej.

Ustawienie danych odbywa się poprzez serię metod zaczynających się słowem "set" i zależnie od pola, które nas interesuje po słowie kluczowym znajduje się nazwa pola.

Pobieranie danych odbywa się analogicznie nazwanymi metodami, gdzie słowo kluczowe "set" jest zastąpione słowem "get".

### **Klasa Kid.java**

Klasa odpowiada za przechowywanie i ustawianie danych dziecka.

Przechowuje dane takie jak:

Imię, Nazwisko, data urodzenia oraz informacje o alergiach.

Możemy sprawdzić dane dziecka przy użyciu metody "checkInformationAboutKid"

### **Klasa Menu.java**

Klasa odpowiada za przechowywanie i ustawianie danych jadłospisu.

Przechowuje dane takie jak:

Typ posiłku oraz z czego się składa.

Możemy sprawdzić dane dziecka przy użyciu metody "checkInformationAboutMenu".

W tej klasie możemy także ustawić jednym razem typ posiłku i jego skład przy użyciu metody

### **Klasa Meeting.java**

Klasa odpowiada za sprawdzanie informacji o spotkaniu przy użyciu metody "checkInformationAboutMeeting".

### **Klasa Organizational.java**

Klasa odpowiada za sprawdzanie informacji o spotkaniu przy użyciu metody "checkInformationAboutMeeting", która w tym przypadku zwraca string "Spotkanie organizacyjne". Implementuje ona także klasę Meeting.java.

### **Klasa Types.java**

Klasa odpowiada za sprawdzanie informacji o spotkaniu przy użyciu metody "checkInformationAboutMeeting".

Klasa ta posiada też pole "meet" które odpowiada za przechowywanie typu spotkania. Implementuje ona także klasę Meeting.java.

### **Klasa Contest.java**

Klasa jest rozszerzeniem klasy Types. Zawiera informacje na temat szczególnego rodzaju spotkania (Meeting) dotyczącego konkursu. Składa się z pól opisujących nazwę, datę, opiekunę, grupę oraz opis powiązane ze spotkaniem.

Metoda checkInformationAboutMeeting() zwraca informacje o spotkaniu Meeting rozszerzone o informacje o spotkaniu ContestMeeting.

Metoda checkInformationAboutContestMeeting() służy sprawdzeniu informacji o spotkaniu dotyczącym konkursu.

### **Klasa EducationalMeeting.java**

Klasa jest rozszerzeniem klasy Types. Zawiera informacje na temat szczególnego rodzaju spotkania (Meeting) – wychowawczego. Składa się z pól opisujących nazwę, datę, opiekunkę, grupę oraz opis powiązane ze spotkaniem. Metoda checkInformationAboutMeeting() zwraca informacje o spotkaniu Meeting rozszerzone o informacje o spotkaniu EducationalMeeting. Metoda checkInformationAboutEducationalMeeting() służy sprawdzeniu informacji o spotkaniu dotyczącym konkursu. Jest to odpowiednik klasy Contest opisujący inny przypadek.

### **Klasa ParentTeacherConference.java**

Klasa ta odpowiada za ustawienie danych dotyczących spotkania z rodzicami, typu data , dla której grupy i dodatkowe dane.

Klasa jest rozszerzeniem "Types.java".

Posiada pola dla daty, do przypisania opiekunki do spotkania, grupy a także opisu.

Ustawienie danych odbywa się poprzez użycie funkcji "ParentTeacherConference".

Jest tutaj także zaimplementowana wcześniej wspomniana metoda

"checkInformationAboutMeeting" w tym wypadku zwracająca dodatkowo wynik "checkInformationAboutParentTeacherConferenceeting", która zwraca dane spotkania wymienione wyżej oraz nazwę spotkania jako "z rodzicami".

### **Klasa Tour.java**

Klasa ta odpowiada za ustawienie danych dotyczących spotkania z rodzicami, typu data , dla której grupy i dodatkowe dane.

Klasa jest rozszerzeniem "Types.java".

Posiada pola dla daty, do przypisania opiekunki do spotkania, grupy a także opisu.

Jest tutaj także zaimplementowana wcześniej wspomniana metoda

"checkInformationAboutMeeting" w tym wypadku zwracająca dodatkowo wynik "checkInformationAboutTourMeeting", która zwraca dane spotkania wymienione wyżej oraz nazwę spotkania jako "w sprawie wycieczki".

Ustawienie danych odbywa się poprzez użycie funkcji "Tour".

### **Klasa Timetable.java**

Klasa ta odpowiada za odczytywanie danych dotyczących grafiku.

Posiada pola instancji dla tworzenia nowego obiektu Timetable oraz listę dla PartOfTimetable.

Jest tutaj zaimplementowana metoda "checkInformationAboutTimetable", która poprzez iteracje sprawdza grafik lub zwraca , że plan jest pusty w przypadku pustego grafiku.

Funkcja getTimetable zwraca zmienną timetable.

### **Klasa PartOfTimetable.java**

Klasa ta odpowiada za ustawienie i odczytywanie danych dotyczących grafiku.

Posiada pola dla dnia, przypisania opiekunki do grafiku, grupy oraz godziny a także pole odpowiadające za numer dnia tygodnia, które jest używane w switch case w celu ustawienia odpowiedniego dnia.

Ustawianie danych odbywa się w funkcji PartOfTimetable.

Jest także zaimplementowana funkcja "toString" zwracająca string zawierający dzień, godzinę, grupę oraz opiekunkę przypisaną do grafiku w tym czasie.

### **Klasa User.java**

Klasa ta odpowiada za ustawienie i odczytywanie danych dotyczących użytkownika.

Posiada pola do ustawienia imienia, nazwiska, loginu oraz hasła użytkownika.

Sprawdzenie informacji o użytkowniku odbywa się poprzez "checkInformationAboutUser".

### **Klasa Babysitter.java**

Klasa jest rozszerzeniem do klasy Employee. Odpowiada za przechowywanie informacji na temat specjalizacji opiekunki. Posiada dwie metody służące do dodawania i pobierania informacji o specjalizacji. Metoda checkInformationAboutBabysitter() służy do sprawdzenia informacji o opiekunce.

### **Klasa Cook.java**

Klasa przechowująca metodę do obsługi Menu – setInformationAboutMenu(). Zawiera informację o posiłku jaki będzie podany dzieciom. Klasa ma także dostęp do informacji o alergenach, ponieważ podczas sporządzania jadłospisu konieczne jest uwzględnienie dzieci mających uczulenie na konkretne produkty spożywcze.

### **Klasa Toys.java**

Klasa przechowuje dane dotyczące zabawek znajdujących się w przedszkolu, tj. typ, numer seryjny, cena. Zawiera metody pozwalające na wprowadzanie zmian w spisie zabawek, dodawanie i usuwanie ich. Metoda setTypeOfToy() służy to ustawienia typu zabawki, np. zabawka może być edukacyjna. Metoda setSerialNumber() odpowiada za numerowanie zabawek, jest to potrzebne, ponieważ może być więcej niż jedna taka sama zabawka. Za dodanie ceny zabawki odpowiada metoda setPrice(). Do pobrania informacji na temat wymienionych parametrów służą odpowiednio metody ze słowem „get” zamiast „set”. Ostatnia metoda checkInformationAboutToy() służy do pobrania kompletu informacji o zabawce.

### **Klasa ForKidsThreePlus.java**

Klasa jest rozszerzeniem klasy Toys. Zawiera informacje na temat zabawek wyłącznie dla dzieci powyżej 3 roku życia. Zawiera pole name przechowujące nazwę zabawki a także metody do ustawiania tej nazwy oraz pobierania jej. Zawiera też metodę checkInformationAboutToy() w rozszerzonej wersji – dodatkowo wyświetli nazwę zabawki.

### **Klasa ForKidsUnderThree.java**

Klasa jest rozszerzeniem klasy Toys. Zawiera informacje na temat zabawek przeznaczonych dla dzieci do 3 roku życia. Przechowuje nazwę zabawki, którą można dodać metodą setNameOfToy() oraz pobrać poprzez getNameOfToy(). Metoda checkInformationAboutToy() wyświetli Typ, numer seryjny, cenę, a także nazwę zabawki.

### **Klasa Principal.java**

Klasa jest rozszerzeniem klasy User. Zawiera metody, do których dostęp może mieć jedynie Dyrektor przedszkola. Przechowuje listę dzieci, pracowników a także zabawek. Metody w tej klasie służą dodawaniu pracowników – addEmployeeToSystem(), usuwaniu – deleteEmployeeFromTheSystem(), sprawdzeniu listy pracowników - checkListOfEmployeesInSystem(). Kolejne metody służą do obsługi listy dzieci, analogicznie są to: addChildToSystem(), deleteChildFromSystem(), checkListOfKidsInSystem(). Każda grupa dzieci ma przypisaną opiekunkę, służy do tego metoda addBabysitterToGroup(). Aby usunąć opiekunkę istnieje metoda deleteBabysitterFromTheGroup(). Metoda odpowiadająca za wprowadzenie do systemu zakupionych zabawek: buyToys(). Dyrektora ma także możliwość wprowadzania zmian w grafiku metodą modifyInformationAboutTimetable() oraz sprawdzenie grafiku poprzez checkInformationAboutTimetable().

### **Klasa Employee.java**

Klasa jest rozszerzeniem klasy User. Zawiera metody do których dostęp ma zarówno Dyrektor jak i opiekunka. Umożliwia utworzenie grupy dzieci oraz dodawanie do niej dzieci spośród tych znajdujących się w systemie. Posiada także metodę służącą sprawdzeniu listy dzieci w systemie checkListOfChildrenInSystem() oraz usunięciu dziecka z grupy. Klasa posiada też metodę checkInformationAboutTimetable() zwracającą informacje o grafiku.

### **Klasa Parent.java**

Klasa jest rozszerzeniem klasy User. Zawiera informacje na temat Indexu dziecka, do którego przypisany jest rodzic. Posiada też pola przechowujące adres i numer kontaktowy do rodzica a także metody pozwalające wprowadzać zmiany w w/w polach. Informacje o rodzicu można sprawdzić poprzez metodę checkInformationAboutParent(). Metoda checkInformationAboutTimetable() służy do sprawdzania informacji o grafiku.

### **Klasa Informative.java**

Klasa jest rozszerzeniem klasy Meeting. Przedstawia szczególny przypadek spotkania – spotkanie informacyjne. Zawiera metodę String checkInformationAboutMeeting() zwracającą typ spotkania (pole private final String name), czyli ciąg znaków „spotkanie informacyjne”.

### **Klasa Events.java**

Klasa zawiera informacje na temat wydarzenia odbywającego się w przedszkolu innego niż codzienne zajęcia i spotkanie z rodzicami. Może dotyczyć np. zabawy choinkowej dla dzieci, wizyty teatru lub dnia sportowego. Klasa przechowuje informacje dotyczące rodzaju wydarzenia, miejsca i czasu

### **Klasa Group.java**

Klasa zawiera informację na temat listy dzieci przypisanych do danej grupy oraz opieunki, która się nimi zajmuje. Każda grupa ma też przypisany swój index w celu ich identyfikacji. Informacje te można sprawdzić metodą checkInformationAboutGroup().

Ponadto klasy Meeting, Informative Meeting, Organizational Meeting, Types, Tour, Contest Educational Meeting, ParentTeacherMeeting zostały stworzone w oparciu o wzorzec strukturalny Dekorator. Klasa Timetable jest Singletonem.

Dodatkowo w projekcie znajdują się klasy testowe w pakiecie TestClasses. Służą one do przeprowadzania testów jednostkowych dla odpowiednich klas (np. TestBabysitter testuje klasę Babysitter).

Wśród plików projektowych jest także pakiet Main zawierający implementację prostego systemu logowania w formie okna logowania, niestety nie zostało to wykorzystane.