

РОССИЙСКИЙ УНИВЕРСИТЕТ ДРУЖБЫ НАРОДОВ

Факультет физико-математических и естественных наук

Кафедра прикладной информатики и теории вероятностей

ОТЧЕТ

ПО ЛАБОРАТОРНОЙ РАБОТЕ №2

дисциплина: Моделирование сложноструктурированных систем

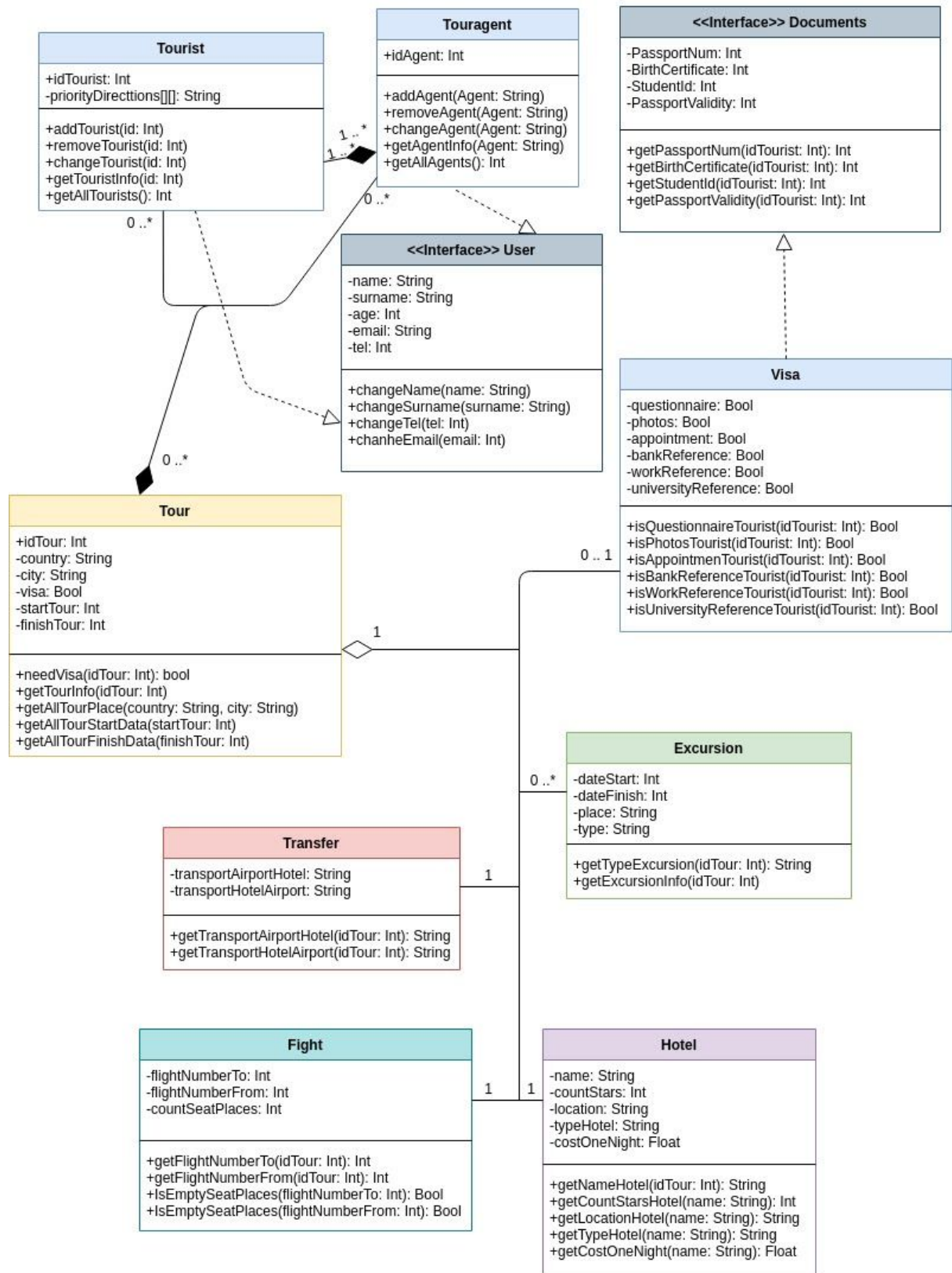
Студент: Уточкина Маргарита

Группа: НПИ-02-17

МОСКВА

2020 г.

<class> Диаграмма классов по турагентству



Описание классификаторов

<<interface>> *User*

Атрибуты:

-name: String	// Имя
-surname: String	// Фамилия
-age: Int	// Возраст
-email: String	// Эл. почта
-tel: Int	// Телефон

Методы:

+changeName(name: String)	// Изменить Имя
+changeSurname(surname: String)	// Изменить фамилию
+changeTel(tel: Int)	// Изменить телефон
+chanheEmail(email: Int)	// Изменить Почту

<<interface>> *Tourist*

Атрибуты:

+idTourist: Int	// id туриста
-priorityDirections[][]: String	// Приоритетные направления для путешествия

Методы:

+addTourist(id: Int)	// добавить туриста
+removeTourist(id: Int)	// удалить туриста
+changeTourist(id: Int)	// изменить данные туриста
+getTouristInfo(id: Int)	// получить информацию о туристе
+getAllTourists(): Int	// получить список всех туристов

<<interface>> Touragent

Атрибуты:

+idAgent: Int // id тур.агента

Методы:

+addAgent(Agent: String) // добавить тур.агента
+removeAgent(Agent: String) // удалить тур.агента
+changeAgent(Agent: String) // изменить данные тур.агента
+getAgentInfo(Agent: String) // информацию о тур.агенте
+getAllAgents(): Int // получить список всех тур.агентов

<<interface>> Tour

Атрибуты:

+idTour: Int // id тура
-country: String // страна посещения
-city: String // город посещения
-visa: Bool // проверка, есть или нет визы
-startTour: Int // дата начала тура
-finishTour: Int // дата окончания тура

Методы:

+needVisa(idTour: Int): bool // проверка, есть или нет визы
+getTourInfo(idTour: Int) // получение информации о туре
+getAllTourPlace(country: String, city: String) // показать все туры по
определенному месту
+getAllTourStartDate(startTour: Int) // показать все туры, начинающиеся с
// определенной даты
+getAllTourFinishData(finishTour: Int) // показать все туры,

// заканчивающиеся в определенную
// дату

<<interface>> *Transfer*

Атрибуты:

-transportAirportHotel: String // информация о
//транспорте от аэропорта до отеля
-transportHotelAirport: String // информация о
//транспорте от отеля до аэропорта

Методы:

+getTransportAirportHotel(idTour: Int): String // показать транспорт от
//аэропорта до отеля
+getTransportHotelAirport(idTour: Int): String // показать транспорт от
//отеля до аэропорта

<<interface>> *Excursion*

Атрибуты:

-dateStart: Int // дата начала экскурсии
-dateFinish: Int // дата окончания экскурсии
-place: String // место экскурсии
-type: String // тип экскурсии

Методы:

+getTypeExcursion(idTour: Int): String // получить тип экскурсии
// по id тура

+getExcursionInfo(idTour: Int) // получить информацию
// об экскурсии по id тура

<<interface>> *Flight*

Атрибуты:

-flightNumberTo: Int // номер рейса туда
-flightNumberFrom: Int // номер рейса обратно
-countSeatPlaces: Int // кол-во мест

Методы:

+getFlightNumberTo(idTour: Int): Int // получить номер рейса туда
+getFlightNumberFrom(idTour: Int): Int // получить номер рейса туда
+IsEmptySeatPlaces(flightNumberTo: Int): Bool // получить кол-во
// свободных мест туда
+IsEmptySeatPlaces(flightNumberFrom: Int): Bool // получить кол-во
// свободных мест обратно

<<interface>> *Hotel*

Атрибуты:

-name: String // название отеля
-countStars: Int // кол-во звезд в отеле
-location: String // месторасположение отеля
-typeHotel: String // тип отеля
-costOneNight: Float // цена за одну ночь проживания

Методы:

+getNameHotel(idTour: Int): String	// получить имя отеля по id тура
+getCountStarsHotel(name: String): Int	// получить кол-во звезд отеля
+getLocationHotel(name: String): String	// получить локацию отеля
+getTypeHotel(name: String): String	// получить тип отеля
+getCostOneNight(name: String): Float	// получить цену за одну ночь проживания

<<interface>> Visa

Атрибуты:

-questionnaire: Bool	// есть ли анкета
-photos: Bool	// есть ли фото
-appointment: Bool	// есть ли заявление
-bankReference: Bool	// есть ли справка из банка
-workReference: Bool	// есть ли справка с работы
-universityReference: Bool	// есть ли справка с учебы

Методы:

+isQuestionnaireTourist(idTourist: Int): Bool	// есть ли анкета у
//определенного туриста - получение этой информации по idTourist	
+isPhotosTourist(idTourist: Int): Bool	// есть ли фото у
//определенного туриста - получение этой информации по idTourist	
+isAppointmentTourist(idTourist: Int): Bool	// есть ли заявление у
//определенного туриста - получение этой информации по idTourist	
+isBankReferenceTourist(idTourist: Int): Bool	// есть ли справка из банка
//у определенного туриста - получение этой информации по idTourist	
+isWorkReferenceTourist(idTourist: Int): Bool	// есть ли справка с работы
//у определенного туриста - получение этой информации по idTourist	
+isUniversityReferenceTourist(idTourist: Int): Bool	// есть ли справка с
//учёбы у определенного туриста - получение этой информации по	
//idTourist	

<<interface>> Documents

Атрибуты:

-PassportNum: Int // номер паспорта
-BirthCertificate: Int // номер свидетельства о рождении
-StudentId: Int // номер studentid
-PassportValidity: Int // срок действия паспорта

Методы:

+getPassportNum(idTourist: Int): Int // получить номер паспорта по idTourist
+getBirthCertificate(idTourist: Int): Int // получить номер номер // свидетельства о рождении по idTourist
+getStudentId(idTourist: Int): Int // получить studentid по idTourist
+getPassportValidity(idTourist: Int): Int // получить срок действия паспорта по idTourist

Описание взаимосвязей между классификаторами

Touragent

Класс Touragent реализует интерфейс User - это показывается с помощью отношения реализации.

Класс Touragent связан с классом Touragent посредством использования связи Композиция (см. объяснение в Tourist). Touragent может быть от 0 до максимально возможного значения int в языке, на котором диаграмма была бы реализована.

Tourist

Класс Tourist реализует интерфейс User - это показывается с помощью отношения реализации.

Класс Tourist связан с классом Touragent посредством использования связи Композиция, что означает, что класс Tourist накладывает определенные требования на класс Touragent и класс Touragent перестает существовать вместе с классом Tourist. Tourist может быть от 0 до максимально возможного значения int в языке, на котором диаграмма была бы реализована.

User

Классификатор (в данном случае интерфейс) User связан с классификаторами Tourist, Touragent с помощью связи Реализация, что означает, что классификаторы (классы в данном случае) имеют те же атрибуты и методы, что описаны в классификаторе User.

Tour

Классификатор Tour связан с классами Tourist, Touragent с помощью использования связи Композиция. Это означает, что классы Tourist, Touragent накладывают некоторые требования на класс Tour. Tour может быть от 0 до максимально возможного значения int в языке, на котором диаграмма была бы реализована.

Классификатор Tour связан с классами Transfer, Flight, Hotel, Excursion, Visa посредством использования связи Агрегация. Это означает, что экземпляры классов Transfer, Flight, Hotel, Excursion, Visa входят в состав экземпляра класса Tour.

Transfer

Класс Transfer связан с Tour с помощью агрегации. Объяснение см. в Tour.

Flight

Класс Flight связан с Tour с помощью агрегации. Объяснение см. в Tour.

Hotel

Класс Hotel связан с Tour с помощью агрегации. Объяснение см. в Tour.

Excursion

Класс Excursion связан с Tour с помощью агрегации. Объяснение см. в Tour.

Visa

Класс Visa связан с Tour с помощью агрегации. Объяснение см. в Tour.

Класс Visa реализует интерфейс Documents – это показывается с помощью отношения реализации.

Documents

Классификатор (в данном случае интерфейс) Documents связан с классификатором Visa с помощью связи Реализация, что означает, что классификаторы (классы в данном случае) имеют те же атрибуты и методы, что описаны в классификаторе Documents.