**ОПИСАНИЕ ОСОБЕННОСТЕЙ РЕАЛИЗАЦИИ КЛИЕНТСКОГО МОДУЛЯ**

**СИСТЕМЫ АВТОМАТИЗИРОВАННОГО СБОРА ДАННЫХ**

**ВЕРСИЯ 1.0**

**СОДЕРЖАНИЕ**

1. ПРИМЕЧАНИЯ 1

2. СТРУКТУРА ПРОЕКТА КЛИЕНТСКОГО МОДУЛЯ 2

3. ОСОБЕННОСТИ РЕАЛИЗАЦИИ И ИСПОЛЬЗОВАНИЯ КЛАССОВ 8

**ПРИМЕЧАНИЯ**

*Здесь могут быть примечания по особенностям форматирования кода, именования переменных и функций, использования комментариев и инструментария документации, возможным направлениям рефакторинга (TODO) и т.д.*

**СТРУКТУРА ПРОЕКТА КЛИЕНТСКОГО МОДУЛЯ**

|  |  |
| --- | --- |
| Client | Класс, представляющий клиента |
| Consultant | Класс, представляющий консультанта |
| Couple | Класс, представляющий пару(создание свидания для пары) |
| Users | Класс, представляющий пользователя(администратор и консультанты – их логины и пароли) |
| Authorization | Класс авторизации |
| PickUpCouple | Класс, отвечающий за подбор кандидатов для клиента |
| Log | Класс логирования |

**Основные классы**

**Методы и свойства класса Client**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Client | | | | |
| Название | | | Описание | |
| Свойства | | | | |
| Additionally | | | | Дополнительные сведения о желаемом партнере для клиента |
| attitudeToBadHabits | | | | Отношение клиента к вредным привычкам |
| availabilityOfCar | | | | Наличие машины у клиента |
| availabilityOfchildren | | | | Наличие детей у клиента |
| bodyType | | | | Телосложение клиента |
| characterTraits | | | | Черты характера клиента |
| children\_sAccommodation | | | | Место проживания детей клиента |
| cntM | | | | Счетчик экземпляров класса(мужчин) |
| cntW | | | | Счетчик экземпляров класса (женщин) |
| code | | | | Код клиента |
| criminalRecord | | | | Судимости клиента |
| dateOfBirth | | | | Дата рождения клиента |
| e\_mail | | | | e-mail клиента |
| education | | | | Образование клиента |
| eyeColor | | | | Цвет глаз клиента |
| financialPosition | | | | Финансовое положение клиента |
| fullName | | | | ФИО клиента |
| fullYears | | | | Количество полных лет клиента |
| hairColor | | | | Цвет волос клиента |
| height | | | | Рост клиента |
| housingConditions | | | | Жилищные условия клиента |
| interestsOrHobbies | | | | Интересы или хобби клиента |
| lifeCredo | | | | Жизненные ценности клиента |
| lifePriorities | | | | Жизненные приоритеты клиента |
| listOfCandidates | | | | Список кандидатов на роль партнера для клиента |
| listOfMeetings | | | | Список встреч клиента |
| maritalStatus | | | | Материальное положение клиента |
| nationality | | | | Национальность клиента |
| notNeeded | | Какой партнер клиенту не нужен | | |
| partner\_sAge | | Возраст желаемого партнера | | |
| partner\_sAttitudeToBadHabits | | Отношение желаемого партнера к вредным привычкам | | |
| partner\_sBodyType | | Телосложение делаемого партнера | | |
| partner\_sChildrenAccommodation | | Проживание детей желаемого партнера | | |
| partner\_sEducation | | Образование желаемого партнера | | |
| partner\_sHeight | | Рост желаемого партнера | | |
| partner\_sHousingConditions | | Жилищные условия желаемого партнера | | |
| partner\_sImportantQualities | | Важные качества желаемого партнера | | |
| partner\_sNationality | | Национальность желаемого партнера | | |
| partner\_sPlaceOfLiving | | Место проживания желаемого партнера | | |
| partner\_sReligion | | Религия желаемого партнера | | |
| partner\_sWeight | | Вес желаемого партнера | | |
| partnerHasChildren | | Наличие детей у желаемого партнера | | |
| phoneNumber | | Номер телефона клиента | | |
| placeOfLiving | | Место проживания клиента | | |
| placeOfWork | | Место работы клиента | | |
| position | | Должность клиента | | |
| registeredInDispensary | | Состоит ли клиент на учете в диспансере | | |
| religion | | Религия клиента | | |
| term | | Период, которой клиент планирует получать услуги агентства | | |
| timeForCommunication | | Удобное время для встречи клиента | | |
| toHaveChildrenInMarriage | | Желание клиента заводить детей в браке | | |
| weight | | Вес клиента | | |
| Методы | | | | |
| Client | Конструктор | | | |
| Client(string[]) | Конструктор, принимающий массив строк, элементами которого заполняет поля класса | | | |
| GetInfo() | Метод для получения информации о клиенте | | | |
| GetReport | Метод для получения информации о работе с клиентом (отчет) | | | |
| Code | Геттер для персонального кода клиента | | | |
| FullName | Геттер и сеттер для ФИО клиента | | | |
| FullYears | Геттер и сеттер для возраста клиента | | | |
| ListOfCandidates | Геттер и сеттер для списка кандидатов клиента | | | |
| ListOfMeetings | Геттер и сеттер для списка встреч клиента | | | |

**Методы и свойства класса Consultant**

|  |  |  |  |
| --- | --- | --- | --- |
| Consultant | | | |
| Название | | Описание | |
| Свойства | | | |
| positionCode | Персональный код консультанта | | |
| fullName | ФИО консультанта | | |
| age | Возраст консультанта | | |
| telephone | Номер телефона консультанта | | |
| numberOfClients | Количество клиентов консультанта | | |
| numberOfPairs | Количество удачно соединенных пар консультанта | | |
| dateOfEmployment | Дата принятия на работу | | |
| lastEmployment | Прошлое место работы | | |
| experience | Опыт работы | | |
| address | Адрес проживания | | |
| passportData | Паспортные данные | | |
| salary | Заработная плата | | |
| cnt | Количество экземпляров класса | | |
| password | Пароль | | |
| listOfClients | Список клиентов консультанта | | |
| Методы | | | |
| FullName | | | Сеттер и геттер ФИО консультанта |
| GetInfo() | | | Метод для получения информации о консультанте |
| GetReport() | | | Метод для получения отчета о работе консультанта |

**Методы и свойства класса Couple**

|  |  |  |  |
| --- | --- | --- | --- |
| Couple | | | |
| Название | | | Описание |
| Свойства | | | |
| firstClientFullName | ФИО первого клиента | | |
| secondClientFullName | ФИО второго клиента | | |
| firstClientCode | Персональный код первого клиента | | |
| secondClientCode | Персональный код второго клиента | | |
| dateOfTheMeeting | Дата для встречи пары | | |
| meetingPlase | Место встречи пары | | |
| cnt | Счетчик всех экземпляров класса | | |
| Методы | | | |
| firstClientFullName | | Сеттер и геттер ФИО первого клиента | |
| secondClientFullName | | Сеттер и геттер ФИО второго клиента | |
| GetInfo() | | Метод для получения информации о паре | |
| Couple | | Конструктор | |

**Методы и свойства класса Users**

|  |  |  |
| --- | --- | --- |
| Users | | |
| Название | | Описание |
| Свойства | | |
| Logins | | Список всех логинов |
| Passwords | | Список всех паролей |
| Методы | | |
| Users | Конструктор | |

**Методы и свойства класса Authorization**

|  |  |  |  |
| --- | --- | --- | --- |
| Authorization | | | |
| Название | | | Описание |
| Свойства | | | |
| Login | | Логин пользователя | |
| password | | Пароль пользователя | |
| user | | пользователь | |
| Методы | | | |
| AddUser | Метод для добавления нового пользователя | | |
| EnterToForm | Метод для проверки данных пользователя | | |
| LoadUsers | Метод для загрузки данных всех пользователей | | |
| Authorization | Конструктор | | |

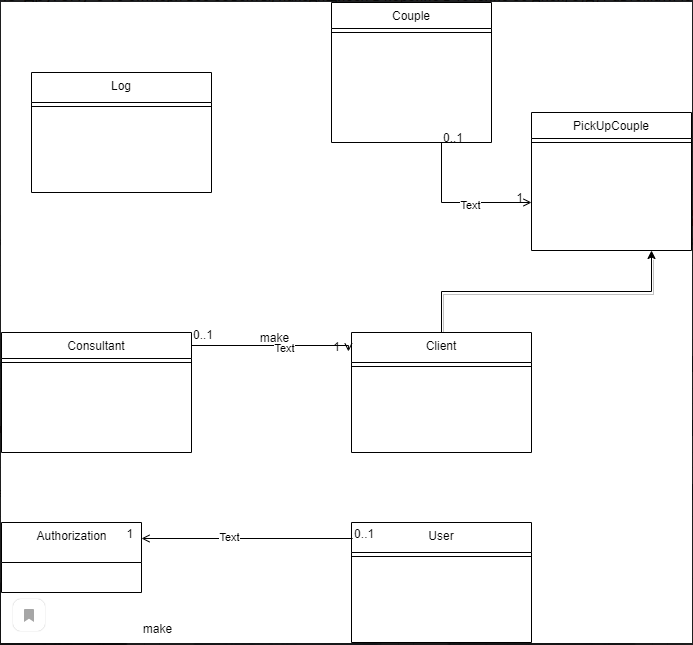
**Методы и свойства класса PickUpCouple**

|  |  |  |
| --- | --- | --- |
| PickUpCouple | | |
| Название | | Описание |
| Свойства | | |
| client | | Клиент, которому подбирают пару |
| Методы | | |
| CreateLisTOfCandidates | Метод для подбора списка кандидатов | |
| PickUpCouple | Метод для подбора пары | |

**Методы и свойства класса Log**

|  |  |  |  |
| --- | --- | --- | --- |
| Log | | | |
| Название | | Описание | |
| Свойства | | | |
| Name | Название | | |
| Operation | Суть операции | | |
| Time | Время операции | | |
| Методы | | | |
| Log | | | конструктор |
| GetLog | | | Метод для получения отчета |

Общая схема классов уровня логики приложения приведена на Рис.1.

****

**Рисунок 1 – UML-диаграмма классов уровня логики приложения**

**ОСОБЕННОСТИ РЕАЛИЗАЦИИ И ИСПОЛЬЗОВАНИЯ КЛАССОВ**

* **Client**

Класс, позволяющий представить в программе объект клиента.

Ниже приведен программный код системы

public class Client

{

public Client()

{ }

public Client(string[] info) { }

public string FullName

{

get => fullName;

}

public string Code

{

get => code;

}

public Dictionary<string, string> ListOfCandidates

{

get => listOfCandidates;

set => listOfCandidates = value;

}

public Dictionary<string, string> ListOfMeetings

{

get => listOfMeetings;

set => listOfMeetings = value;

}

public byte FullYears

{

get => fullYears;

}

string[] GetInfo() { return null; }

string GetReport() { return null; }

private string code;

private string fullName;

private DateTime dateOfBirth;

private byte fullYears;

private string education;

private string placeOfWork;

private string position;

private decimal financialPosition;

private string maritalStatus;

private bool availabilityOfchildren;

private string children\_sAccommodation;

private string housingConditions;

private bool availabilityOfCar;

private string attitudeToBadHabits;

private byte height;

private byte weight;

private string bodyType;

private string eyeColor;

private string hairColor;

private string religion;

private string nationality;

private string interestsOrHobbies;

private string characterTraits;

private string lifePriorities;

private string lifeCredo;

private string phoneNumber;

private string e\_mail;

private string timeForCommunication;

private string placeOfLiving;

private bool registeredInDispensary;

private bool criminalRecord;

private bool toHaveChildrenInMarriage;

private DateTime term;

private byte partner\_sAge;

private byte partner\_sHeight;

private byte partner\_sWeight;

private string partner\_sBodyType;

private string partner\_sEducation;

private string partner\_sPlaceOfLiving;

private bool partnerHasChildren;

private string partner\_sChildrenAccommodation;

private string partner\_sHousingConditions;

private string partner\_sReligion;

private string partner\_sNationality;

private string partner\_sAttitudeToBadHabits;

private string partner\_sImportantQualities;

private string notNeeded;

private string additionally;

private Dictionary<string, string> listOfCandidates;

private Dictionary<string, string> listOfMeetings;

private static ushort cntW;

private static ushort cntM;

}

* **Consultant**

Класс, позволяющий представить в программе объект консультанта.

Ниже приведен программный код системы

public class Consultant

{

public Consultant()

{

}

private string positionCode;

private string fullName;

private byte age;

private string telephone;

private ushort numberOfClients;

private ushort numberOfPairs;

private DateTime dateOfEmployment;

private string lastEmployment;

private byte experience;

private string address;

private string passportData;

private decimal salary;

private static ushort cnt;

private string password;

private List<Client> listOfClients;

public string FullName

{

get => fullName;

}

string[] GetInfo()

{

return null;

}

string GetReport()

{

return null;

}

}

* **Couple**

Класс, позволяющий в программе представить объект пары – создать свидание для двух клиентов, тем самым создав пару.

Ниже приведен программный код системы

public class Couple

{

public Couple()

{

}

private string firstClientFullName;

private string secondClientFullName;

private string firstClientCode;

private string secondClientCode;

private DateTime dateOfTheMeeting;

private string meetingPlase;

private static ushort cnt;

public string FirstClientFullName

{

get => firstClientFullName;

}

public string SecondClientFullName

{

get => secondClientFullName;

}

string[] GetInfo()

{

return null;

}

}

* **Users**

Класс, представляющий пользователя (администратор и консультанты – их логины и пароли).

Ниже приведен программный код системы

public class Users

{

public Users()

{

}

List<string> Logins;

List<string> Passwords;

}

* **Authorization**

Класс обеспечивающий процесс авторизации пользователя (администратора или консультанта).

Ниже приведен программный код системы

public class Authorization

{

public string login;

public string password;

private Users user;

public Authorization()

{

}

private void LoadUsers()

{

}

private void EnterToForm()

{

}

private void AddUser()

{

}

}

* **PickUpCouple**

Класс, позволяющий осуществлять подбор кандидатов для клиента.

Ниже приведен программный код системы

public class PickUpCouple

{

public PickUpCouple (Client client, List<Client> listOfClient)

{ }

public Client client;

public List<Client> CreateLisTOfCandidates()

{ return null; }

}

}

* **Log**

Класс, позволяющий логировать все события и действия пользователя.

Ниже приведен программный код системы

public class Log

{

public Log()

{

}

private DateTime time;

private string name;

private string operation;

public void GetLog()

{ }

}

}